Douglas FLI106xx Family

PRELIMINARY Datasheet

Ordering Information

Part Number	Output Resolution	Package	Temperature Range
FLI10610H/HM[1]-AB	WSXGA+ (1680 x 1050) WXGA+ (1440 x 900P) WXGA (1366 x 768)	633 PBGA	TBD
FLI10620H/HM-AB	1080p (1920 x 1080)	633 PBGA	TBD
FLI10636H/HM-AB	WUXGA (1920 x 1200)	633 PBGA	TBD

Note (1): Some versions require Macrovision license agreement.
Note (2): Macrovision enabled parts are only sold to Macrovision licensed customers.

C106XX-DAT-01E

April 17, 2008

Documentation Feedback

Genesis Microchip Inc. recognizes the need for accurate and easy to use documentation; in case of further clarification, your Genesis FAE acts as the first line of contact. For questions concerning this document, please e-mail us at docfeedback@gnss.com with the following details:

- Document title
- Product number
- Document number
- Page number
- Section heading
- Figure or Table number, etc.

These details will make it easier for us to answer your questions better.

REVISION HISTORY

Document	Author	Description	Date
C106XX-DAT-01A	Tan Dadurian/ Dennis Castleman/ Michael Bekerman	Preliminary draft. Initial Revision.	July 28, 2007
C106XX-DAT-01B	Tan Dadurian/ Michael Bekerman/ Alex Lutsenko	Updates to VXI/VXO, VCR Out, etc.	August 3, 2007
C106XX-DAT-01C	Patrick Sun/ Tan Dadurian	Updated Power Sequencing Updated Signal Descriptions: Table 14, Two-Wire, Table 15, LBADC, Table 17, UART, and Table 23, Power & Ground Removed DDR1 references from Memory Controller section Updated Block Diagram Removed 2.5V DDR references Updated CDI Interface diagrams Updated Table 40, PIP Matrix Updated Table 48, Maximum Ratings, Table 49, DC Characteristics, and Table 50, AC Characteristics Added WUXGA part (FLI10636)	February 21, 2008
C106XX-DAT-01D	Tan Dadurian/ Sheena Shi	Added new Macrovision patent numbers to trademarks page Moved pin A26 to No Connect, Table 22	March 5, 2008
C106XX-DAT-01E	Tan Dadurian/ Patrick Sun/ Sheena Shi	Changed pin D26 to CVSS Added Reminder note below Absolute Maximum Ratings table	April 17, 2008

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

1 OVERVIEW

Genesis Microchip's Douglas series ICs are single-chip TV solutions for products requiring superior 10-bit video quality in the analog and/or digital TV for ATSC, DVB, DTMB, NTSC, OpenCable™, and PAL markets. They include a single channel HD MPEG2 decoder and a secondary Picture-in-Picture (PIP) channel, flexible analog front end with an integrated Faroudja® 3D Video Decoder, high performance industry standard 32-bit MIPS 4Kec processor (250 MIPS), multi-standard analog audio decoder, digital audio decoder and post processor, three programmable Multimedia Processing Engines (MPEs), advanced 2D graphics engine, integrated HDMI 1.3/DVI receivers with integrated HDCP support, and a unified DDR memory controller. The Douglas series chips also include a very flexible and unique Video eXpansion Interface (VXI) providing glueless connectivity to Genesis video processors or a customer's proprietary video processing. All the video processing on Douglas is 10-bit for enhanced video quality of Main and PIP channels and is the first integrated DTV controller to showcase Faroudja RealColor™.

All family members include the next generation Faroudja DCDi Cinema video format conversion, video enhancement, and noise reduction. The level of video quality that could previously only be seen on an exclusive Faroudja Home Theater System is now available in a single-chip solution.

	FLI10610	**FLI10620**	**FLI10636**
	Douglas-Plus	**Douglas-Advanced**	**Douglas-WUXGA**
Max Display Resolution	WSXGA+ (1680x1050) WXGA+ (1440 x 900) WXGA (1366 x 768)	1080p (1920 x 1080)	WUXGA (1920 x 1200)
Graphics	1680x1050 @ 8 bpp 1440 x 900 @ 16/32 bpp	1920 x 1080 @ 8 bpp 1366 x 768 @ 16 bpp	1920 x 1200 @ 8 bpp
Package	633 BGA 37mm x 37mm		
10-bit LVDS	2x (WXGA+ 120Hz / 1080p 60Hz / WUXGA 60Hz)		
HDMI/HDCP	2x		
CI/CableCARD I/F	Yes		
DDR Memory	x32, 360MHz		
VXI	Yes		
NOR/NAND Flash	Yes		
LBADC Input	6 channels		
Audio Out	Five Reconfigurable Audio DACs + SPDIF + Four I^2S		
Audio Input	5 Stereo inputs + SPDIF + I^2S		
Analog Video Capture	16 user definable video inputs + 4 fixed inputs		
Picture-in-Picture (PIP)	All inputs may be multiplexed to either channel		

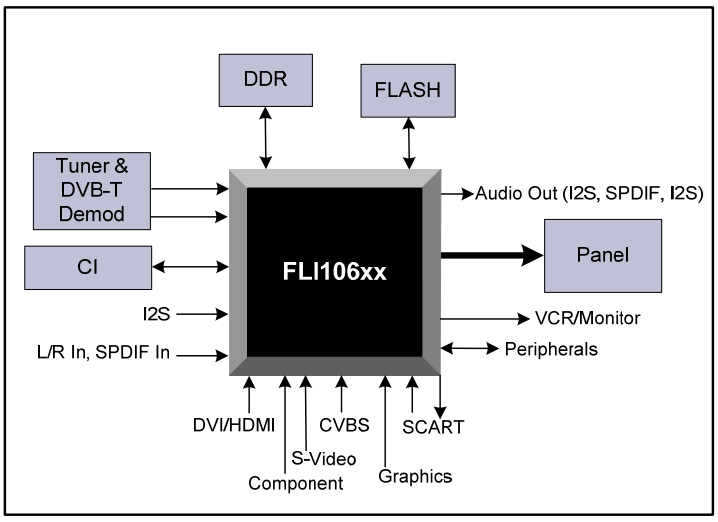


Figure 1. FLI106xx System Design

The FLI106xx chips consist of the following major functional blocks, as shown in Figure 2. These blocks are described briefly below.

Compressed Data Interface (CDI):

- Provides a transport/program stream input interface configurable serial input with dedicated clock recovery logic (parallel input is supported in CableCard™ mode)
- Supports an almost glueless interface with CableCARD/Common Interface connectors

Compressed Data Output (CDO):

- Outputs TS data in serial mode to the external decoder

Transport/Program Stream De-Multiplexing:

- Supports MPEG2 MP@ML, MP@HL (ISO13818-1) transport stream
- De-multiplexes and PID filters OpenCable™, ATSC, and DVB transport streams
- Supports up to 64 PID and 128 section filters
- Provides an integrated DES-ECB descrambler for OpenCable applications

Audio Decoding and Post-Processing:

- Class A certified Dolby® Digital (AC3) decode, including emergency broadcast
- Also supports the decode of MPEG1 Layer I/II, MPEG2 Layer II and MP3
- Supports synchronization of audio (decoded internally or externally) with video
- Capable of supporting audio post processing algorithms like Dolby Prologic, SRS TruSurround XT™, Equalizer, audio delay insertion, and Bass redirection

Video Decoding:

- Single MPEG2 MP@HL (ISO13818-2) video decoding as constrained by ATSC, OpenCable, and DVB standards
- Error detection and concealment at all levels including slice and macroblock
- Capable of decoding MPEG1 (ISO11172-2), JPEG, and other emerging video compression standards

System Processor (MIPS):

- Provides an integrated 32-bit MIPS 4Kec™ processor (250 MIPS)
- Capable of supporting all major operating systems including Linux®

Memory Interface (DRAM Controller):

- Provides a Unified Memory Architecture that supports multiple clients
- X16 – 400 MHz or x32 – 333MHz interface for JEDEC compliant DDR2 DRAMs
- Supports total DDR memory size from 16MB to 128MB (64MB x 2)

2D Graphics Subsystem (DE2D):

- Provides fast memory-to-memory copy for blitting and scrolling
- Supports horizontal line draws, color keying, color fill, block copy, and 16 Raster Operations (ROPS)
- Supports up to 32bpp aRGB with 256 levels of alpha blending or 8bpp LUT
- Supports color conversion and mono-chrome to color expansion
- Two hardware graphic planes with scaling support for UI, MHEG-5, MHP, and Subtitles

Faroudja DCDi Cinema® Format Conversion:

- Low Angle De-interlacing Processing
- Per Pixel Motion Adaptive De-interlacing (MADi) for input sources up to 1080i format
- Format conversion for input and display resolutions up to 1080p
- Panoramic and anamorphic non-linear scaling
- Adaptive Media Display Processing for 3:2 and 2:2 video content
- Adaptive 3D Noise Reduction and Noise Coring
- Media Noise Reduction for MPEG inputs

Faroudja TrueLife™ Video Enhancer:

- Non-linear Chroma and Luma enhancement
- High performance programmable sharpening filters with Noise Coring
- Removal of DVD Chroma Upsampling Error (CUE) introduced by some DVDs

VBI Signal Processing:

- Multi-standard Digital VBI dataslicer
- WST Teletext FastText page support access
- V-chip, VPS, Closed Captioning, XDS, CGMS, and WSS decode
- CC and digital teletext VBI re-insertion

Integrated 3D Video Decoder:

- Faroudja IntelliComb™ Technology
- 3D Adaptive Comb Filter for Luma–Chroma separation
- Composite, S-Video, and Component (including High Definition) video input
- Supports all broadcast TV Video standards—NTSC (North America and Japan), PAL (I, B, G, H, M, D, N), SECAM (D, K, L, B, G)
- Macrovision® and VCR trick mode support

Integrated HDMI 1.3/DVI 1.0 Capture and 2→1 HDMI MUX:

- Dual MUX with HDMI 1.3/DVI input ports (Link and Phy)
- 1080p HDMI capture support
- Integrated HDCP 1.3 key storage
- xvYCC, Deep Color, and Wide Gamut support

Digital and Analog Video Capture:

- Direct connection from tuner or connector to analog input pin
- Three dedicated RGB pins for high speed VGA performance
- One dedicated SIF input and one shared SIF input (from Video inputs)
- 16 additional reconfigurable analog video inputs of which one can be SIF input
- RGB or YUV capture up to 165 MHz
- Full SCART support including RGB Fast Blank
- Supports quadruple 10-bit ADCs for better SNR with one ADC dedicated for SIF

Peripherals Connectivity:

- Video eXpansion Interface – One 16-bit 4:2:2 YCbCr interface output and one 24-bit TTL digital video input port
- Interface to Genesis MCTi co-processor for motion compensated frame interpolation
- Interfaces to third-party H.264 decode/PVR co-processors
- Configurable LVDS/TTL panel interface port (up to 1080p, single-wide TTL, single/double-wide LVDS)
- Integrated USB 2.0 Host Link and Phy, UARTs, Master and Slave 2-wire interfaces, IR receiver, PWMs, and multiple GPIOs
- Flexible host bus interface that enables NOR & NAND Flash support
- SPI FLASH interface
- Integrated NTSC/PAL video encoder and four 10-bit video DACs
- Direct CableCARD/CI interface
- Low-bandwidth ADC

Audio Interfaces:

- Analog Sound IF, Analog Stereo, I2S and SPDIF audio inputs; integrated 5 audio DACs, SPDIF and I2S audio outputs
- Supports one 5.1 and two stereo I2S format audio out interface
  - Supports sampling frequencies up to 96 KHz and resolution up to 32 bits
- Supports one SPDIF audio out interface (IEC 60958/61937) and SPDIF audio inputs
- Provides one I2S audio input port for connecting to BTSC decoders, Karaoke input, etc.
  - Supports sampling frequencies up to 96 KHz and resolution up to 32 bits

Advanced Color Management:

- Advanced Color Management with overlapping regions allows flexible Flesh-Tone compensation, Blue Stretch, color regions detection, and other image enhancements
- Faroudja RealColor provides flexible programming, polar coordinate representation, and independent six-axis color control
- Advanced Contrast Control delivers smoother, more realistic gradients and ensures that full dynamic range is used in video content
- Patented QuickMatch technology produces uniform color responses for different panels using flexible and programmable techniques

Debug Support:

- Provides software debug interface via the EJTAG port
- Supports industry standard debuggers, such as X-Ray (from Mentor) and GDB
- Supports JTAG for IEEE 1149 boundary scan

Clocks and Power Management:

- All clocks are derived from a single external 19.6 MHz crystal
- Advanced block-by-block power management to reduce power consumption to meet ENERGY STAR® requirements

Packaging:

- 633 Plastic Ball Grid Array Package (PBGA)

Power Supplies:

- Core VDD: 1.2V
- DDR I/F:
  - DDR2 operation VDD = 1.8V
- IO VDD: 3.3V

2 FUNCTIONAL DESCRIPTION

2.1 FLI106XX BLOCK DIAGRAM

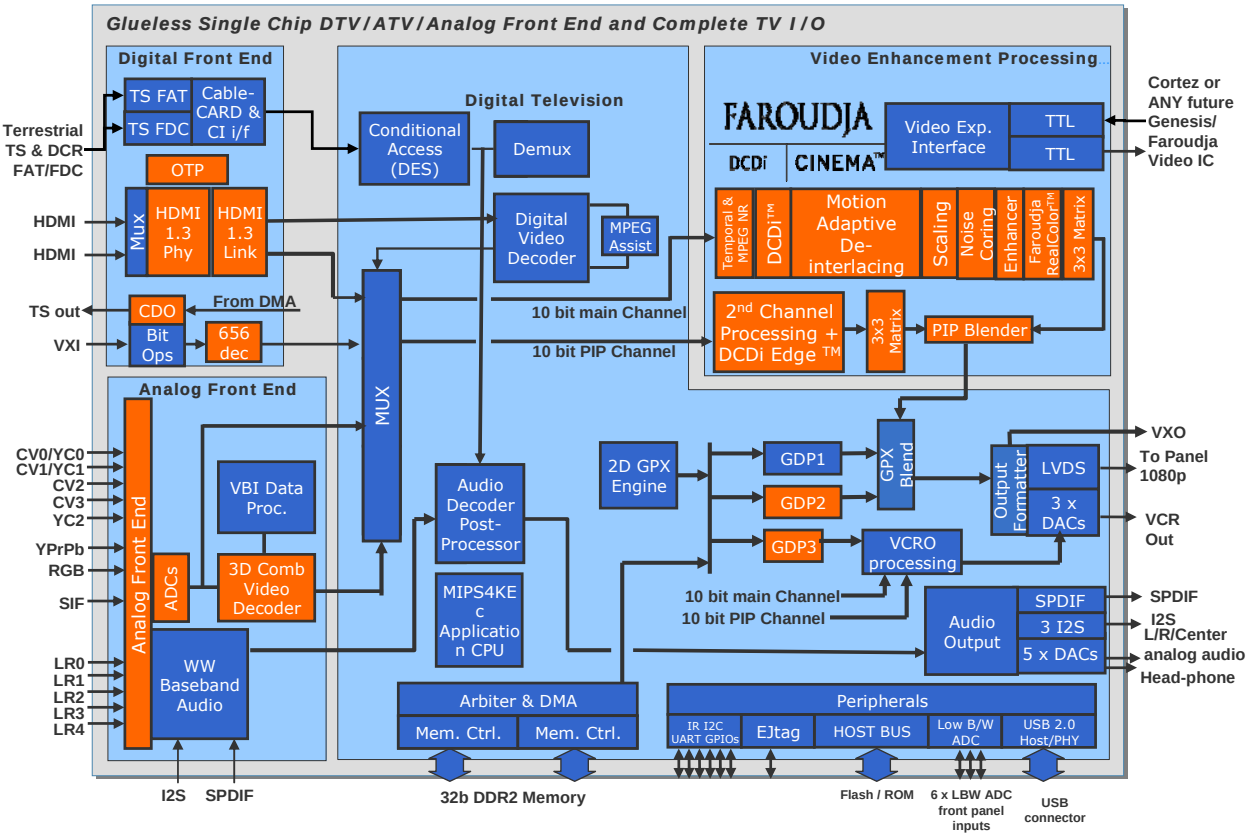


Figure 2. Block Diagram

3 PIN DETAILS

The FLI106xx chip is available in a 633 pin PBGA (Plastic Ball Grid Array) package. Figure 3 depicts the pin location of all the FLI106xx signals.

3.1 FLI106XX PIN DIAGRAM (FULL)

A	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	B
A	ADC_VDD12	HDMI_GNDA	BRX2M	BRX1M	BRX0M	BRXCM	ARX2M	ARX1M	ARX0M	ARXCM	HDMI_CEC	2WIRE_S0_SCL	PWM3	VXI_D17	VXI_D12	VXI_D6	VXI_D1	VXI_CLK	UART0_RXD	UART1_RXD	PPWR	REF_CLK	XTAL_IN	LBADC_IN6	LBADC_IN5		TDO	TMS	EJ_RST_N	A
B	B1P	ADC_VDD12	BRX2P	BRX1P	BRX0P	BRXCP	ARX2P	ARX1P	ARX0P	ARXCP	HDMI_A_HPD	2WIRE_S0_SDA	PWM2	VXI_D18	VXI_D13	VXI_D7	VXI_D2	VXI_VS	UART0_TXD	UART1_TXD	PBIAS	RPLL_AGND	RPLL_AVDD33	LBADC_IN4	LBADC_IN3	TRST	TDI	TCK	EJ_DINT	B
C	C1P	A1P	ADC_VDD12	HDMI_VDDA33	HDMI_GNDA	HDMI_GNDA	HDMI_GNDA	HDMI_GNDA	HDMI_GNDA	HDMI_GNDA	HDMI_B_HPD	2WIRE_S1_SCL	PWM1	VXI_D19	VXI_D14	VXI_D8	VXI_D3	VXI_DE	UART1_CTS	UART1_RTS	IRDATA	RPLL_AGND	RPLL_AVDD12	LBADC_IN2	LBADC_IN1	TESTMODE1	TESTMODE0	USB_PWREN	USB_FLAG	C
D	B2P	AN	SV1P	ADC_GND12	HDMI_VDDA33	HDMI_VDDA33	HDMI_GNDA	HDMI_GNDA	REXT	HDMI_VDD12	2WIRE_S2_SCL	2WIRE_S1_SDA	PWM0	VXI_D20	VXI_D15	VXI_D9	VXI_D4	VXI_HS	VXI_D10	2WIRE_M1_SCL	2WIRE_M1_SDA	DDRPLL_AVDD12	RESET_N	LBADC_33	DDR_VDDQ	CVSS	DDR_VDDQ	DDR_VSSO	DDR_VDDQ	D
E	C2P	A2P	ADC_GNDA	ADC_VDDA33	ADC_GND12	HDMI_VDDA33	HDMI_VDDA33	HDMI_VDDA33	HDMI_VDD12	HDMI_VDD12	2WIRE_S2_SDA	CLKOUT	DFSYNC	VXI_D21	VXI_D16	VXI_D11	VXI_D22	VXI_D23	VXI_D0	2WIRE_M0_SCL	DDRPLL_AGND	DDRPLL_AVDD33	LBADC_GND	DDR_VSSO	DDR_D8	DDR_VSSO	DDR_D13	DDR_VSSO	DDR_D9	E
F	ADC_GNDA	BN	SV2P	ADC_VDDA33	ADC_GNDA	ADC_GND12					IOVDD33	IOVDD33	IOVDD33	IOVDD33	IOVDD33	IOVDD33	VXI_D5	IOVDD33	IOVDD33	2WIRE_M0_SDA	OBUFC_CLK	LBADC_RETURN	DDR_VDDI	DDR_D0	DDR_VDDQ	DDR_D7	DDR_VDDQ	DDR_D15	DDR_VDDQ	F
G	B3P	A3P	ADC_GNDA	ADC_VDDA33	ADC_GNDA	SCART_FB																		DDR_VSSO	DDR_VREF_0	DDR_VSSO	DDR_VSSO	DDR_DQS1	DDR_DQS1_N	G
H	C3P	CN	SV3P	ADC_VDDA33	ADC_GNDA	AVS																		DDR_VDDQ	DDR_DQS0	DDR_DQS0_N	DDR_VDDQ	DDR_D10	DDR_VDDQ	H
J	G_GRA	B_GRA	SV4P	VOUT2	ADC_GNDA	AHS_ACS																		DDR_VSSO	DDR_D2	DDR_VSSO	DDR_DM1	DDR_VSSO	DDR_D11	J
K	R_GRA	SVN	A4P	ADC_VDDA33	ADC_GNDA	SIF_IN																		DDR_DM0	DDR_VDDQ	DDR_D3	DDR_VDDQ	DDR_D14	DDR_VDDQ	K
L	C4P	B4P	AUD_VREFP	AUD_VREFN	AUD_VCM	AUD_MONO_IN					CVDD12	CVDD12	CVSS	CVSS	CVSS	CVSS	CVDD12	CVDD12	CVDD12					DDR_VSSO	DDR_D6	DDR_VSSO	DDR_D12	DDR_VSSO	DDR_D9	L
M	AUD_IN_L2	AUD_IN_R2	AUD_IN_L1	AUD_IN_R1	AUD_IN_R3	SIF_RTN					CVDD12	CVDD12	CVSS	CVSS	CVSS	CVSS	CVDD12	CVDD12	CVDD12					DDR_D1	DDR_VDDQ	DDR_D4	DDR_ODT	DLL_Vaa	DDR_VDDQ	M
N	AUD_IN_L5	AUD_IN_R5	AUD_IN_L4	AUD_IN_R4	AUD_IN_L3	AUD_AVSS12					CVSS	CVSS	CVSS	CVSS	CVSS	CVSS	CVSS	CVSS	CVSS					DDR_VDDQ	DDR_CS_N	DDR_RAS_N	DDR_VSSO	DDR_VSSO	DDR_CK_N	N
P	AUD_AVSS33	AUD_AVSS33	AUD_AVSS33	AUD_AVSS33	AUD_AVSS33	AUD_AVDD12					CVSS	CVSS	CVSS	CVSS	CVSS	CVSS	CVSS	CVSS	CVSS					DDR_CAS_N	DDR_A4	DDR_VSSO	DDR_A0	DDR_VDDQ	DDR_CK	P
R	AUD_OUT1_L	AUD_OUT1_R	LS_OUT_L	LS_OUT_R	LS_OUT_SW	AUD_AVDD12					CVSS	CVSS	CVSS	CVSS	CVSS	CVSS	CVSS	CVSS	CVSS					DDR_VREF_1	DDR_VDDQ	DDR_A2	DDR_A6	DDR_A11	DDR_A8	R
T	AUD_OUT_HP_L	AUD_OUT_HP_R	AUD_OUT2_L	AUD_OUT2_R	AUD_AVDD33	AUD_AVDD33					CVSS	CVSS	CVSS	CVSS	CVSS	CVSS	CVSS	CVSS	CVSS					DDR_VSSO	DDR_WE_N	DDR_VSSO	DDR_BA1	DDR_A5	DDR_A9	T
U	AUD_HP_AVSS33	AUD_HP_AVDD33	AUD_AVDD33	AUD_AVDD33	AUD_AVDD33	IOVDD33					CVSS	CVSS	CVSS	CVSS	CVSS	CVSS	CVSS	CVSS	CVSS					DDR_CAL	DDR_VDDQ	DDR_A3	DDR_CKE	DDR_BA0	DDR_A1	U
V	POD_RESET	POD_READY_IRQ_N	POD_WAIT_N	POD_CD_2	POD_CD_1	IOVDD33					CVDD12	CVDD12	CVDD12	CVSS	CVSS	CVSS	CVSS	CVDD12	CVDD12					DLL_Vaa	DDR_A10	DDR_A12	DDR_VDDQ	DDR_A7	DDR_VSSO	V
W	OOB_CTX	OOB_CRX	OOB_DRX	POD_CE_2	POD_A14_MCLKO	POD_VS2_MCLKO					CVDD12	CVDD12	CVDD12	CVSS	CVSS	CVSS	CVSS	CVDD12	CVDD12					DDR_VSSO	DDR_D21	DDR_VSSO	DDR_D29	DDR_VSSO	DDR_D24	W
Y	POD_D13_MDO5	POD_D10_MDO2	POD_D8_MDO0	POD_BVD1_MOSTRT	POD_BVD2_MOVAL	IOVDD33																		DDR_D16	DDR_VDDQ	DDR_D23	DDR_VDDQ	DDR_D31	DDR_VDDQ	Y
AA	POD_D14_MDO6	POD_D11_MDO3	POD_D9_MDO1	AUDIN_I2S_DAT	AUDIN_SPDIF_IN	IOVDD33																		DDR_VSSO	DDR_VREF_2	DDR_VSSO	DDR_VSSO	DDR_DQS3	DDR_DQS3_N	AA
AB	POD_D15_MDO7	POD_D12_MDO4	AUDIN_I2S_WCLK	AUDO_I2SA_BCLK	AUD_MCLK1	IOVDD33																		DDR_VDDQ	DDR_DQS2	DDR_DQS2_N	DDR_VDDQ	DDR_D26	DDR_VDDQ	AB
AC	AUD_MCLK0	AUDIN_I2S_BCLK	AUDO_I2SB_BCLK	AUDO_I2SA_DAT0	AUDO_I2SA_WCLK	VDAC_VSS12																		DDR_VSSO	DDR_D18	DDR_VSSO	DDR_DM3	DDR_VSSO	DDR_D27	AC
AD	AUDO_SPDIF_OUT	AUDO_I2SB_DAT1	AUDO_I2SB_WCLK	POD_DETECT_N	POD_DIR_N	VDAC_VDD12	VDAC_AVSS33	IOVDD33	HOST_A24	IOVDD33	IOVDD33	IOVDD33	HOST_A21	IOVDD33	VXO_D14	IOVDD33	IOVDD33	IOVDD33	VXO_CLK	USB_GND	LVTX_VSS	LVTX_VSS	DDR_VDDI	DDR_DM2	DDR_VDDQ	DDR_D19	DDR_VDDQ	DDR_D30	DDR_VDDQ	AD
AE	AUDO_MUTE	AUDO_I2SD_DAT2	POD_A4_CTX	POD_CE_1	HOST_DEV_CS2_N	VDAC_AVDD33	POD_HOST_D3	HOST_D10	HOST_D13	HOST_A17	POD_HOST_A13	HOST_A20	HOST_A8	POD_HOST_A3	VXO_D3	VXO_D4	VXO_D9	VXO_D15	VXO_DE	USB_AVDD12	LVTX_VDD33	LVTX_VDD33	LVTX_VSS	DDR_VSSO	DDR_D2	DDR_VSSO	DDR_D28	DDR_VSSO	DDR_D25	AE
AF	POD_A8_CRX	POD_A6_ETX	POD_A5_ITX	HOST_DEV_CS1_N	VDAC_AVDD33	VDAC_AVDD33	HOST_D8	POD_OE_HOST_RD	POD_HOST_D5	HOST_D15	HOST_A14	HOST_A9	HOST_A18	PODREG_HOST_A4	VXO_D0	VXO_D5	VXO_D10	VXO_HS	USB_GND	USB_AVDD33	LVTX_EVN_CH4N_DISP1	LVTX_EVN_CLKN_DISP7	LVTX_EVN_CH1N_DISP11	LVTX_VDD33	DDR_VDDQ	DDR_D20	DDR_VDDQ	DDR_D17	DDR_VDDQ	AF
AG	POD_A9_DRX	POD_A7_QTX	HOST_ACK	VDAC_AVSS33	VDAC_COMP	HOST_BOOT_CS_N	POD_HOST_D0	POD_HOST_D2	HOST_D12	POD_HOST_D7	HOST_A15	POD_HOST_A10	HOST_A19	POD_IOWR_HOST_A5	POD_HOST_A2	VXO_D6	VXO_D11	VXO_VS	USB_AVDD33	USBPHY_VRES	LVTX_EVN_CH4P_DISP0	LVTX_EVN_CLKP_DISP6	LVTX_EVN_CH1P_DISP10	LVTX_EVN_CH5P_DISPNS	LVTX_EVN_CH5N_DISPVS	LVTX_PLL_VDD33	LVTX_VSS	LVTX_ODD_CH5P_DISP0E	LVTX_ODD_CH5N_DISPCLK	AG
AH	VDAC_BU_N	VDAC_RV_N	VDAC_GY_YC_N	VDAC_RSET	HOST_READY	CDI0_VALID	CDI0_CLK	HOST_D9	POD_HOST_D4	HOST_D14	HOST_A16	POD_HOST_A11	HOST_A22	POD_IORD_HOST_A6	POD_HOST_A1	VXO_D1	VXO_D7	VXO_D12	USB_AVDD33	USBPHY_PADM	LVTX_EVN_CH3N_DISP5	LVTX_EVN_CH2N_DISP9	LVTX_EVN_CH3N_DISP13	LVTX_ODD_CH4N_DISP3	LVTX_ODD_CLKN_DISP15	LVTX_ODD_CH4N_DISP17	LVTX_ODD_CH2N_DISP19	LVTX_ODD_CH1N_DISP21	LVTX_ODD_CH0N_DISP23	AH
AJ	VDAC_BU_P	VDAC_RV_P	VDAC_GY_YC_P	HOST_DEV_CS0_N	CDI0_ERROR	CDI0_SYNC	CDI0_D0	POD_HOST_D1	HOST_D11	POD_HOST_D6	HOST_A23	POD_HOST_A12	POD_WE_HOST_WR	HOST_A7	POD_HOST_A0	VXO_D2	VXO_D8	VXO_D13	USB_GND	USBPHY_PADP	LVTX_EVN_CH3P_DISP4	LVTX_EVN_CH2P_DISP8	LVTX_EVN_CH0P_DISP12	LVTX_ODD_CH4P_DISP2	LVTX_ODD_CH3P_DISP14	LVTX_ODD_CLKP_DISP16	LVTX_ODD_CH2P_DISP18	LVTX_ODD_CH1P_DISP20	LVTX_ODD_CH0P_DISP22	AJ
C	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	D

Figure 3. FLI106xx Pin Diagram

3.2 FLI106XX PIN DIAGRAM (ZOOM-IN)

A	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
A	ADC_VDD12	HDMI_GNDA	BRX2M	BRX1M	BRX0M	BRXCM	ARX2M	ARX1M	ARX0M	ARXCM	HDMI_CEC	2WIRE_S0_SCL	PWM3	VXI_D17	VXI_D12
B	B1P	ADC_VDD12	BRX2P	BRX1P	BRX0P	BRXCP	ARX2P	ARX1P	ARX0P	ARXCP	HDMI_A_HPD	2WIRE_S0_SDA	PWM2	VXI_D18	VXI_D13
C	C1P	A1P	ADC_VDD12	HDMI_VDDA33	HDMI_GNDA	HDMI_GNDA	HDMI_GNDA	HDMI_GNDA	HDMI_GNDA	HDMI_GNDA	HDMI_B_HPD	2WIRE_S1_SCL	PWM1	VXI_D19	VXI_D14
D	B2P	AN	SV1P	ADC_GND12	HDMI_VDDA33	HDMI_VDDA33	HDMI_GNDA	HDMI_GNDA	REXT	HDMI_VDD12	2WIRE_S2_SCL	2WIRE_S1_SDA	PWM0	VXI_D20	VXI_D15
E	C2P	A2P	ADC_GNDA	ADC_VDDA33	ADC_GND12	HDMI_VDDA33	HDMI_VDDA33	HDMI_VDD12	HDMI_VDD12	HDMI_VDD12	2WIRE_S2_SDA	CLKOUT	DFSYNC	VXI_D21	VXI_D16
F	ADC_GNDA	BN	SV2P	ADC_VDDA33	ADC_GNDA	ADC_GND12						IOVDD33	IOVDD33	IOVDD33	IOVDD33
G	B3P	A3P	ADC_GNDA	ADC_VDDA33	ADC_GNDA	SCART_FB									
H	C3P	CN	SV3P	ADC_VDDA33	ADC_GNDA	AVS									
J	G_GRA	B_GRA	SV4P	VOUT2	ADC_GNDA	AHS_ACS									
K	R_GRA	SVN	A4P	ADC_VDDA33	ADC_GNDA	SIF_IN									
L	C4P	B4P	AUD_VREFP	AUD_VREFN	AUD_VCM	AUD_MONO_IN					CVDD12	CVDD12	CVSS	CVSS	CVSS
M	AUD_IN_L2	AUD_IN_R2	AUD_IN_L1	AUD_IN_R1	AUD_IN_R3	SIF_RTN					CVDD12	CVDD12	CVSS	CVSS	CVSS
N	AUD_IN_L5	AUD_IN_R5	AUD_IN_L4	AUD_IN_R4	AUD_IN_L3	AUD_AVSS12					CVSS	CVSS	CVSS	CVSS	CVSS
P	AUD_AVSS33	AUD_AVSS33	AUD_AVSS33	AUD_AVSS33	AUD_AVSS33	AUD_AVDD12					CVSS	CVSS	CVSS	CVSS	CVSS
R	AUD_OUT1_L	AUD_OUT1_R	LS_OUT_L	LS_OUT_R	LS_OUT_SW	AUD_AVDD12					CVSS	CVSS	CVSS	CVSS	CVSS

Figure 4. FLI106xx Pin Diagram (Section-A)

16	17	18	19	20	21	22	23	24	25	26	27	28	29	B
VXI_D6	VXI_D1	VXI_CLK	UART0_RXD	UART1_RXD	PPWR	REF_CLK	XTAL_IN	LBADC_IN6	LBADC_IN5		TDO	TMS	EJ_RST_N	A
VXI_D7	VXI_D2	VXI_VS	UART0_TXD	UART1_TXD	PBIAS	RPLL_AGND	RPLL_AVDD33	LBADC_IN4	LBADC_IN3	TRST	TDI	TCK	EJ_DINT	B
VXI_D8	VXI_D3	VXI_DE	UART1_CTS	UART1_RTS	IRDATA	RPLL_AGND	RPLL_AVDD12	LBADC_IN2	LBADC_IN1	TESTMODE1	TESTMODE0	USB_PWREN	USB_FLAG	C
VXI_D9	VXI_D4	VXI_HS	VXI_D10	2WIRE_M1_SCL	2WIRE_M1_SDA	DDRPLL_AVDD12	RESET_N	LBADC_33	DDR_VDDQ	CVSS	DDR_VDDQ	DDR_VSSO	DDR_VDDQ	D
VXI_D11	VXI_D22	VXI_D23	VXI_D0	2WIRE_M0_SCL	DDRPLL_AGND	DDRPLL_AVDD33	LBADC_GND	DDR_VSSO	DDR_D5	DDR_VSSO	DDR_D13	DDR_VSSO	DDR_D8	E
IOVDD33	VXI_D5	IOVDD33	IOVDD33	2WIRE_M0_SDA	OBUFC_CLK	LBADC_RETURN	DDR_VDDI	DDR_D0	DDR_VDDQ	DDR_D7	DDR_VDDQ	DDR_D15	DDR_VDDQ	F
								DDR_VSSO	DDR_VREF_0	DDR_VSSO	DDR_VSSO	DDR_DQS1	DDR_DQS1_N	G
								DDR_VDDQ	DDR_DQS0	DDR_DQS0_N	DDR_VDDQ	DDR_D10	DDR_VDDQ	H
								DDR_VSSO	DDR_D2	DDR_VSSO	DDR_DM1	DDR_VSSO	DDR_D11	J
								DDR_DM0	DDR_VDDQ	DDR_D3	DDR_VDDQ	DDR_D14	DDR_VDDQ	K
CVSS	CVDD12	CVDD12	CVDD12					DDR_VSSO	DDR_D6	DDR_VSSO	DDR_D12	DDR_VSSO	DDR_D9	L
CVSS	CVDD12	CVDD12	CVDD12					DDR_D1	DDR_VDDQ	DDR_D4	DDR_ODT	DLL_Vaa	DDR_VDDQ	M
CVSS	CVSS	CVSS	CVSS					DDR_VDDQ	DDR_CS_N	DDR_RAS_N	DDR_VSSO	DDR_VSSO	DDR_CK_N	N
CVSS	CVSS	CVSS	CVSS					DDR_CAS_N	DDR_A4	DDR_VSSO	DDR_A0	DDR_VDDQ	DDR_CK	P
CVSS	CVSS	CVSS	CVSS					DDR_VREF_1	DDR_VDDQ	DDR_A2	DDR_A6	DDR_A11	DDR_A8	R

Figure 5. FLI106xx Pin Diagram (Section-B)

T	AUD_OUT_HP_L	AUD_OUT_HP_R	AUD_OUT2_L	AUD_OUT2_R	AUD_AVDD33	AUD_AVDD33					CVSS	CVSS	CVSS	CVSS	CVSS
U	AUD_HP_AVSS33	AUD_HP_AVDD33	AUD_AVDD33	AUD_AVDD33	AUD_AVDD33	IOVDD33					CVSS	CVSS	CVSS	CVSS	CVSS
V	POD_RESET	POD_READY_IRQ_N	POD_WAIT_N	POD_CD_2	POD_CD_1	IOVDD33					CVDD12	CVDD12	CVDD12	CVSS	CVSS
W	OOB_CTX	OOB_CRX	OOB_DRX	POD_CE_2	POD_A14_MCLKO	POD_VS2_MCLKO					CVDD12	CVDD12	CVDD12	CVSS	CVSS
Y	POD_D13_MDO5	POD_D10_MDO2	POD_D8_MDO0	POD_BVD1_MOSTRT	POD_BVD2_MOVAL	IOVDD33									
AA	POD_D14_MDO6	POD_D11_MDO3	POD_D9_MDO1	AUDIN_I2S_DAT	AUDIN_SPDIF_IN	IOVDD33									
AB	POD_D15_MDO7	POD_D12_MDO4	AUDIN_I2S_WCLK	AUDO_I2SA_BCLK	AUD_MCLK1	IOVDD33									
AC	AUD_MCLK0	AUDIN_I2S_BCLK	AUDO_I2SB_BCLK	AUDO_I2SA_DAT0	AUDO_I2SA_WCLK	VDAC_VSS12									
AD	AUDO_SPDIF_OUT	AUDO_I2SB_DAT1	AUDO_I2SB_WCLK	POD_DETECT_N	POD_DIR_N	VDAC_VDD12	VDAC_AVSS33	IOVDD33	HOST_A24	IOVDD33	IOVDD33	IOVDD33	HOST_A21	IOVDD33	VXO_D14
AE	AUDO_MUTE	AUDO_I2SB_DAT2	POD_A4_CTX	POD_CE_1	HOST_DEV_CS2_N	VDAC_AVDD33	POD_HOST_D3	HOST_D10	HOST_D13	HOST_A17	POD_HOST_A13	HOST_A20	HOST_A8	POD_HOST_A3	VXO_D3
AF	POD_A8_CRX	POD_A6_ETX	POD_A5_ITX	HOST_DEV_CS1_N	VDAC_AVDD33	VDAC_AVDD33	HOST_D8	POD_OE_HOST_RD	POD_HOST_D5	HOST_D15	HOST_A14	HOST_A9	HOST_A18	PODREG_HOST_A4	VXO_D0
AG	POD_A9_DRX	POD_A7_QTX	HOST_ACK	VDAC_AVSS33	VDAC_COMP	HOST_BOOT_CS_N	POD_HOST_D0	POD_HOST_D2	HOST_D12	POD_HOST_D7	HOST_A15	POD_HOST_A10	HOST_A19	POD_IOWR_HOST_A5	POD_HOST_A2
AH	VDAC_BU_N	VDAC_RV_N	VDAC_GY_YC_N	VDAC_RSET	HOST_READY	CDI0_VALID	CDI0_CLK	HOST_D9	POD_HOST_D4	HOST_D14	HOST_A16	POD_HOST_A11	HOST_A22	POD_IORD_HOST_A6	POD_HOST_A1
AJ	VDAC_BU_P	VDAC_RV_P	VDAC_GY_YC_P	HOST_DEV_CS0_N	CDI0_ERROR	CDI0_SYNC	CDI0_D0	POD_HOST_D1	HOST_D11	POD_HOST_D6	HOST_A23	POD_HOST_A12	POD_WE_HOST_WR	HOST_A7	POD_HOST_A0
C	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15

Figure 6. FLI106xx Pin Diagram (Section-C)

CVSS	CVSS	CVSS	CVSS					DDR_VSSO	DDR_WE_N	DDR_VSSO	DDR_BA1	DDR_A5	DDR_A9	T
CVSS	CVSS	CVSS	CVSS					DDR_CAL	DDR_VDDQ	DDR_A3	DDR_CKE	DDR_BA0	DDR_A1	U
CVSS	CVSS	CVDD12	CVDD12					DLL_Vaa	DDR_A10	DDR_A12	DDR_VDDQ	DDR_A7	DDR_VSSO	V
CVSS	CVSS	CVDD12	CVDD12					DDR_VSSO	DDR_D21	DDR_VSSO	DDR_D29	DDR_VSSO	DDR_D24	W
								DDR_D16	DDR_VDDQ	DDR_D23	DDR_VDDQ	DDR_D31	DDR_VDDQ	Y
								DDR_VSSO	DDR_VRF_2	DDR_VSSO	DDR_VSSO	DDR_DQS3	DDR_DQS3_N	AA
								DDR_VDDQ	DDR_DQS2	DDR_DQS2_N	DDR_VDDQ	DDR_D26	DDR_VDDQ	AB
								DDR_VSSO	DDR_D18	DDR_VSSO	DDR_DM3	DDR_VSSO	DDR_D27	AC
IOVDD33	IOVDD33	IOVDD33	VXO_CLK	USB_GND	LVTX_VSS	LVTX_VSS	DDR_VDDI	DDR_DM2	DDR_VDDQ	DDR_D19	DDR_VDDQ	DDR_D30	DDR_VDDQ	AD
VXO_D4	VXO_D9	VXO_D15	VXO_DE	USB_AVDD12	LVTX_VDD33	LVTX_VDD33	LVTX_VSS	DDR_VSSO	DDR_D22	DDR_VSSO	DDR_D28	DDR_VSSO	DDR_D25	AE
VXO_D5	VXO_D10	VXO_HS	USB_GND	USB_AVDD33	LVTX_EVN_CH4N_DISP1	LVTX_EVN_CLKN_DISP7	LVTX_EVN_CH1N_DISP11	LVTX_VDD33	DDR_VDDQ	DDR_D20	DDR_VDDQ	DDR_D17	DDR_VDDQ	AF
VXO_D6	VXO_D11	VXO_VS	USB_AVDD33	USBPHY_VRES	LVTX_EVN_CH4P_DISP0	LVTX_EVN_CLKP_DISP6	LVTX_EVN_CH1P_DISP10	LVTX_EVN_CH5P_DISPHS	LVTX_EVN_CH5N_DISPVS	LVTX_PLL_VDD33	LVTX_VSS	LVTX_ODD_CH5P_DISPDE	LVTX_ODD_CH5N_DISPCLK	AG
VXO_D1	VXO_D7	VXO_D12	USB_AVDD33	USBPHY_PADM	LVTX_EVN_CH3N_DISP5	LVTX_EVN_CH2N_DISP9	LVTX_EVN_CH0N_DISP13	LVTX_ODD_CH4N_DISP3	LVTX_ODD_CH3N_DISP15	LVTX_ODD_CLKN_DISP17	LVTX_ODD_CH2N_DISP19	LVTX_ODD_CH1N_DISP21	LVTX_ODD_CH0N_DISP23	AH
VXO_D2	VXO_D8	VXO_D13	USB_GND	USBPHY_PADP	LVTX_EVN_CH3P_DISP4	LVTX_EVN_CH2P_DISP8	LVTX_EVN_CH0P_DISP12	LVTX_ODD_CH4P_DISP2	LVTX_ODD_CH3P_DISP14	LVTX_ODD_CLKP_DISP16	LVTX_ODD_CH2P_DISP18	LVTX_ODD_CH1P_DISP20	LVTX_ODD_CH0P_DISP22	AJ
16	17	18	19	20	21	22	23	24	25	26	27	28	29	D

Figure 7. FLI106xx Pin Diagram (Section-D)

3.3 SIGNAL DESCRIPTIONS

I/O Legend: A = Analog, **I** = Input, **O** = Output, **P** = Power, **G**= Ground

Table 1. Clocks

Signal	Ball No	IO Type	Description	Alternate Function
REF_CLK	A22	AI	Crystal Out or external oscillator in	
XTAL_IN	A23	AI	Crystal In	
CLKOUT	E12	IO	Internal clock out	GPIO11
OBUFC_CLK	F21	IO	Test pad: PLL control voltage	

Table 2. CDI

Signal	Ball No	IO Type	Description	Alternate Function
CDI0_ERROR	AJ5	I	Transport Stream Error input	GPI12
CDI0_VALID	AH6	I	Transport Stream Valid input	GPI2
CDI0_SYNC	AJ6	I	Transport Stream Sync byte indication	GPI4
CDI0_CLK	AH7	I	Transport Stream Clock input	GPI3
CDI0_D0	AJ7	I	Transport Stream serial Data input	GPI9

Table 3. DDR

Signal	Ball No	IO Type	Description	Alternate Function
DDR_A0	P27	O	DDR_A0	
DDR_A1	U29	O	DDR_A1	
DDR_A2	R26	O	DDR_A2	
DDR_A3	U26	O	DDR_A3	
DDR_A4	P25	O	DDR_A4	
DDR_A5	T28	O	DDR_A5	
DDR_A6	R27	O	DDR_A6	
DDR_A7	V28	O	DDR_A7	
DDR_A8	R29	O	DDR_A8	
DDR_A9	T29	O	DDR_A9	
DDR_A10	V25	O	DDR_A10	
DDR_A11	R28	O	DDR_A11	
DDR_A12	V26	O	DDR_A12	
DDR_D0	F24	IO	DDR_D0	
DDR_D1	M24	IO	DDR_D1	
DDR_D2	J25	IO	DDR_D2	
DDR_D3	K26	IO	DDR_D3	
DDR_D4	M26	IO	DDR_D4	
DDR_D5	E25	IO	DDR_D5	
DDR_D6	L25	IO	DDR_D6	
DDR_D7	F26	IO	DDR_D7	
DDR_D8	E29	IO	DDR_D8	
DDR_D9	L29	IO	DDR_D9	
DDR_D10	H28	IO	DDR_D10	
DDR_D11	J29	IO	DDR_D11	

Signal	Ball No	IO Type	Description	Alternate Function
DDR_D12	L27	IO	DDR_D12	
DDR_D13	E27	IO	DDR_D13	
DDR_D14	K28	IO	DDR_D14	
DDR_D15	F28	IO	DDR_D15	
DDR_D16	Y24	IO	DDR_D16	
DDR_D17	AF28	IO	DDR_D17	
DDR_D18	AC25	IO	DDR_D18	
DDR_D19	AD26	IO	DDR_D19	
DDR_D20	AF26	IO	DDR_D20	
DDR_D21	W25	IO	DDR_D21	
DDR_D22	AE25	IO	DDR_D22	
DDR_D23	Y26	IO	DDR_D23	
DDR_D24	W29	IO	DDR_D24	
DDR_D25	AE29	IO	DDR_D25	
DDR_D26	AB28	IO	DDR_D26	
DDR_D27	AC29	IO	DDR_D27	
DDR_D28	AE27	IO	DDR_D28	
DDR_D29	W27	IO	DDR_D29	
DDR_D30	AD28	IO	DDR_D30	
DDR_D31	Y28	IO	DDR_D31	
DDR_BA0	U28	O	DDR_BA0	
DDR_BA1	T27	O	DDR_BA1	
DDR_CAL	U24	O	DDR termination calibration pin; connect a 294 ohm +/-1% resistor from this pin to ground.	
DDR_CAS_N	P24	O	DDR_CAS_N	
DDR_CK	P29	O	DDR_CK	
DDR_CK_N	N29	O	DDR_CK_N	
DDR_CKE	U27	O	DDR_CKE	
DDR_CS_N	N25	O	DDR_CS_N chip select output to dram	
DDR_DM0	K24	O	DDR_DM0 data mask	
DDR_DM1	J27	O	DDR_DM1 data mask	
DDR_DM2	AD24	O	DDR_DM2	
DDR_DM3	AC27	O	DDR_DM3	
DDR_DQS0	H25	IO	DDR_DQS0	
DDR_DQS0_N	H26	IO	DDR_DQS0_N	
DDR_DQS1	G28	IO	DDR_DQS1	
DDR_DQS1_N	G29	IO	DDR_DQS1_N	
DDR_DQS2	AB25	IO	DDR_DQS2	
DDR_DQS2_N	AB26	IO	DDR_DQS2_N	
DDR_DQS3	AA28	IO	DDR_DQS3	
DDR_DQS3_N	AA29	IO	DDR_DQS3_N	
DDR_ODT	M27	O	DDR_ODT on die termination control output to dram	
DDR_RAS_N	N26	O	DDR_RAS_N	
DDR_WE_N	T25	O	DDR_WE_N	

Table 4. Analog Front End

Signal	Ball No	IO Type	Description	Alternate Function
A1P	C2	AI	Positive analog input 'A' for channel 1.	
A2P	E2	AI	Positive analog input 'A' for channel 2.	
A3P	G2	AI	Positive analog input 'A' for channel 3.	
A4P	K3	AI	Positive analog input 'A' for channel 4.	
B1P	B1	AI	Positive analog input 'B' for channel 1.	
B2P	D1	AI	Positive analog input 'B' for channel 2.	
B3P	G1	AI	Positive analog input 'B' for channel 3.	
B4P	L2	AI	Positive analog input 'B' for channel 4.	
C1P	C1	AI	Positive analog input 'C' for channel 1.	
C2P	E1	AI	Positive analog input 'C' for channel 2.	
C3P	H1	AI	Positive analog input 'C' for channel 3.	
C4P	L1	AI	Positive analog input 'C' for channel 4.	
AN	D2	AI	Negative analog input 'A' for channels 1 through 4.	
BN	F2	AI	Negative analog input 'B' for channels 1 through 4.	
CN	H2	AI	Negative analog input 'C' for channels 1 through 4.	
AHS_ACS	J6	I	AFE Hsync or Csync input	GPI5
AVS	H6	I	AFE Vsync input	GPI6
SCART_FB	G6	I	AFE SCART Fast Blank input (digital input monitored by AIP_SIG_MON to validate valid fast blank signal).	GPI10
SV1P	D3	AI	Positive analog input or sync input for channel 1.	
SV2P	F3	AI	Positive analog input or sync input for channel 2.	
SV3P	H3	AI	Positive analog input or sync input for channel 3.	
SV4P	J3	AI	Positive analog input or sync input for channel 4.	
SVN	K2	AI	Negative analog input or sync input for channels 1 through 4.	
VOUT2	J4	AO	Analog VOUT signal.	
R_GRA	K1	AI		
G_GRA	J1	AI		
B_GRA	J2	AI		

Table 5. VXI

Signal	Ball No	IO Type	Description	Alternate Function
VXI_D0	E19	I	Video expansion input interface: luminance data.	GPIO72
VXI_D1	A17	I	Video expansion input interface: luminance data.	GPIO73
VXI_D2	B17	I	Video expansion input interface: luminance data.	GPIO74
VXI_D3	C17	I	Video expansion input interface: luminance data.	GPIO75
VXI_D4	D17	I	Video expansion input interface:	GPIO76

Signal	Ball No	IO Type	Description	Alternate Function
			luminance data.	
VXI_D5	F17	I	Video expansion input interface: luminance data.	GPIO77
VXI_D6	A16	I	Video expansion input interface: luminance data.	GPIO78
VXI_D7	B16	I	Video expansion input interface: luminance data.	GPIO79
VXI_D8	C16	I	Video expansion input interface: chroma data.	GPIO80
VXI_D9	D16	I	Video expansion input interface: chroma data.	GPIO81
VXI_D10	D19	I	Video expansion input interface: chroma data.	GPIO82
VXI_D11	E16	I	Video expansion input interface: chroma data.	GPIO83
VXI_D12	A15	I	Video expansion input interface: chroma data.	GPIO84
VXI_D13	B15	I	Video expansion input interface: chroma data.	GPIO85
VXI_D14	C15	I	Video expansion input interface: chroma data.	GPIO86
VXI_D15	D15	I	Video expansion input interface: chroma data.	GPIO87
VXI_D16	E15	I	Video expansion input interface: chroma data.	GPIO88
VXI_D17	A14	I	Video expansion input interface: chroma data.	GPIO89
VXI_D18	B14	I	Video expansion input interface: chroma data.	GPIO90
VXI_D19	C14	I	Video expansion input interface: chroma data.	GPIO91
VXI_D20	D14	I	Video expansion input interface: chroma data.	GPIO92
VXI_D21	E14	I	Video expansion input interface: chroma data.	GPIO93
VXI_D22	E17	I	Video expansion input interface: chroma data.	GPIO94
VXI_D23	E18	I	Video expansion input interface: chroma data.	GPIO95
VXI_CLK	A18	I	Video expansion input interface: Clock.	GPIO68
VXI_DE	C18	I	Video expansion input interface: Display Enable (DE).	GPIO69
VXI_HS	D18	I	Video expansion input interface: HSYNC.	GPIO71
VXI_VS	B18	I	Video expansion input interface: VSYNC.	GPIO70

Table 6. VXO

Signal	Ball No	IO Type	Description	Alternate Function
VXO_D0	AF15	O	Video expansion interface data out: G/Y[0]	GPIO132
VXO_D1	AH16	O	Video expansion interface data out: G/Y[1]	GPIO133
VXO_D2	AJ16	O	Video expansion interface data out: G/Y[2]	GPIO134
VXO_D3	AE15	O	Video expansion interface data out: G/Y[3]	GPIO135
VXO_D4	AE16	O	Video expansion interface data out: G/Y[4]	GPIO136
VXO_D5	AF16	O	Video expansion interface data out: G/Y[5]	GPIO137
VXO_D6	AG16	O	Video expansion interface data out: G/Y[6]	GPIO138
VXO_D7	AH17	O	Video expansion interface data out: G/Y[7]	GPIO139
VXO_D8	AJ17	O	Video expansion interface data out: B/UV/U[0]	GPIO140
VXO_D9	AE17	O	Video expansion interface data out: B/UV/U[1]	GPIO141
VXO_D10	AF17	O	Video expansion interface data out: B/UV/U[2]	GPIO142
VXO_D11	AG17	O	Video expansion interface data out:B/UV/U[3]	GPIO143
VXO_D12	AH18	O	Video expansion interface data out: B/UV/U[4].	GPIO144
VXO_D13	AJ18	O	Video expansion interface data out: B/UV/U[5].	GPIO145
VXO_D14	AD15	O	Video expansion interface data out: B/UV/U[6]	GPIO146
VXO_D15	AE18	O	Video expansion interface data out: B/UV/U[7]	GPIO147
VXO_CLK	AD19	O	Video expansion output interface: Clock.	GPIO128
VXO_DE	AE19	O	Video expansion output interface: Display Enable (DE); can optionally be programmed to be field ID signal (ODD or proscan = 1; Even = 0).	GPIO129
VXO_HS	AF18	O	Video expansion output interface: HSYNC.	GPIO131
VXO_VS	AG18	O	Video expansion output interface: VSYNC. Also GPIO130.	GPIO130

Table 7. LVDS/TTL Transmitter

Signal	Ball No	IO Type	Description	Alternate Function
LVTX_EVN_CH4P_DISP0	AG21	AO	LVDS [even channel 4 +ve], TTL [digital BLU/U bit 0 output]	GPIO127
LVTX_EVN_CH4N_DISP1	AF21	AO	LVDS [even channel 4 –ve], TTL [digital BLU/U bit 1 output]	GPIO126
LVTX_ODD_CH4P_DISP2	AJ24	AO	LVDS [ODD channel 4 +ve], TTL [digital Blu/U bit 2 output]	GPIO115

Signal	Ball No	IO Type	Description	Alternate Function
LVTX_ODD_CH4N_DISP3	AH24	AO	LVDS [ODD channel 4 –ve], TTL [digital Blu/U bit 3 output]	GPIO114
LVTX_EVN_CH3P_DISP4	AJ21	AO	LVDS [even channel 3 +ve], TTL [digital BLU/U bit 4 output]	GPIO125
LVTX_EVN_CH3N_DISP5	AH21	AO	LVDS [even channel 3 –ve], TTL [digital BLU/U bit 5 output]	GPIO124
LVTX_EVN_CLKP_DISP6	AG22	AO	LVDS [even CLK +ve], TTL [digital BLU/U bit 6 output]	GPIO123
LVTX_EVN_CLKN_DISP7	AF22	AO	LVDS [even CLK –ve], TTL [digital BLU/U bit 7 output]	GPIO122
LVTX_EVN_CH2P_DISP8	AJ22	AO	LVDS [even channel 2 +ve], TTL [digital GRN/Y bit 0 output]	GPIO121
LVTX_EVN_CH2N_DISP9	AH22	AO	LVDS [even channel 2 –ve], TTL [digital GRN/Y bit 1 output]	GPIO120
LVTX_EVN_CH1P_DISP10	AG23	AO	LVDS [even channel 1 +ve], TTL [digital GRN/Y bit 2 output]	GPIO119
LVTX_EVN_CH1N_DISP11	AF23	AO	LVDS [even channel 1 –ve], TTL [digital GRN/Y bit 3 output]	GPIO118
LVTX_EVN_CH0P_DISP12	AJ23	AO	LVDS [even channel 0 +ve], TTL [digital GRN/Y bit 4 output]	GPIO117
LVTX_EVN_CH0N_DISP13	AH23	AO	LVDS [even channel 0 –ve], TTL [digital GRN/Y bit 5 output]	GPIO116
LVTX_ODD_CH3P_DISP14	AJ25	AO	LVDS [ODD channel 3 +ve], TTL [digital GRN/Y bit 6 output]	GPIO113
LVTX_ODD_CH3N_DISP15	AH25	AO	LVDS [ODD channel 3 –ve], TTL [digital GRN/Y bit 7 output]	GPIO112
LVTX_ODD_CLKP_DISP16	AJ26	AO	LVDS [ODD CLK +ve], TTL [digital RED/V bit 0 output]	GPIO111
LVTX_ODD_CLKN_DISP17	AH26	AO	LVDS [ODD CLK –ve], TTL [digital RED/V bit 1 output]	GPIO110
LVTX_ODD_CH2P_DISP18	AJ27	AO	LVDS [ODD channel 2 +ve], TTL [digital RED/V bit 2 output]	GPIO109
LVTX_ODD_CH2N_DISP19	AH27	AO	LVDS [ODD channel 2 –ve], TTL [digital RED/V bit 3 output]	GPIO108
LVTX_ODD_CH1P_DISP20	AJ28	AO	LVDS [ODD channel 1 +ve], TTL [digital RED/V bit 4 output]	GPIO107
LVTX_ODD_CH1N_DISP21	AH28	AO	LVDS [ODD channel 1 –ve], TTL [digital RED/V bit 5 output]	GPIO106
LVTX_ODD_CH0P_DISP22	AJ29	AO	LVDS [ODD channel 0 +ve], TTL [digital RED/V bit 6 output]	GPIO105
LVTX_ODD_CH0N_DISP23	AH29	AO	LVDS [ODD channel 0 –ve], TTL [digital RED/V bit 7 output].	GPIO104
LVTX_ODD_CH5N_DISPCLK	AG29	AO	LVDS [ODD channel 5 –ve], TTL [digital Disp. CLK output]	GPIO12
LVTX_ODD_CH5P_DISPDE	AG28	AO	LVDS [ODD channel 5 +ve], TTL [digital DE output]	GPIO13
LVTX_EVN_CH5P_DISPHS	AG24	AO	LVDS [even channel 5 pos.], TTL [digital Hsync output]	GPIO15
LVTX_EVN_CH5N_DISPVS	AG25	AO	LVDS [even channel 5 neg.], TTL [digital Vsync output]	GPIO14

Table 8. DAC

Signal	Ball No	IO Type	Description	Alternate Function
VDAC_BU_N	AH1	AG	Analog return channel for VDAC_BU_P. Connect to pcb analog ground.	
VDAC_BU_P	AJ1	AO	DAC Blue(RGB component), Pb(YprPb component), Chroma (S-Video) or C (CVBS) analog positive output	
VDAC_COMP	AG5	AI	DAC compensation pin input [what were earlier pins by name COL CRS and other pins in v0.2 meant for]	
VDAC_GY_YC_N	AH3	AI	Analog return channel for VDAC_GY_YC_P. Connect to pcb analog ground.	
VDAC_GY_YC_P	AJ3	AO	DAC Green(RGB component), Y(YprPb component), Y (S-Video) or C (CVBS) analog positive output	
VDAC_RSET	AH4	AI	DAC external resistor input	
VDAC_RV_N	AH2	AG	Analog return channel for VDAC_RV_P. Connect to pcb analog ground.	
VDAC_RV_P	AJ2	AO	DAC Red(RGB component), Pr(YprPb component), Y (S-Video) or C (CVBS) analog positive output	

Table 9. Audio In

Signal	Ball No	IO Type	Description	Alternate Function
AUDIN_I2S_BCLK	AC2	I	Audio In I2S Bit Clock	GPIO29
AUDIN_I2S_DAT	AA4	I	Audio In I2S Data	GPIO31
AUDIN_I2S_WCLK	AB3	I	Audio In I2S Word Clock	GPIO30
AUDIN_SPDIF_IN	AA5	I	Audio In SPDIF Data	GPIO32
AUD_IN_L1	M3	AI	Audio analog input #1 left channel	
AUD_IN_R1	M4	AI	Audio analog input #1 right channel	
AUD_IN_L2	M1	AI	Audio analog input #2 left channel	
AUD_IN_R2	M2	AI	Audio analog input #2 right channel	
AUD_IN_L3	N5	AI	Audio analog input #3 left channel	
AUD_IN_R3	M5	AI	Audio analog input #3 right channel	
AUD_IN_L4	N3	AI	Audio analog input #4 left channel	
AUD_IN_R4	N4	AI	Audio analog input #4 right channel	
AUD_IN_L5	N1	AI	Audio analog input #5 left channel	
AUD_IN_R5	N2	AI	Audio analog input #5 right channel	
AUD_MCLK0	AC1	IO	External master clock input/output for Audio	GPIO28
AUD_MCLK1	AB5	IO	External master clock input/output for Audio	GPIO33
AUD_MONO_IN	L6	AI	Karaoke mono input	
SIF_IN	K6	AI		
SIF_RTN	M6	AI		

Table 10. Audio Out

Signal	Ball No	IO Type	Description	Alternate Function
AUDO_I2SA_BCLK	AB4	O	I2S Stereo Channel Output Bit Clock for Port A	GPIO34
AUDO_I2SB_BCLK	AC3	O	I2S Stereo Channel Output Bit Clock for Port B	GPIO37
AUDO_I2SA_DAT0	AC4	O	I2S Stereo Channel Output Data for Port A	GPIO36
AUDO_I2SB_DAT1	AD2	O	I2S Stereo Channel Output Data for Port B	GPIO39
AUDO_I2SB_DAT2	AE2	O	I2S Stereo Channel Output Data for Port B	GPIO40
AUDO_I2SA_WCLK	AC5	O	I2S Stereo Channel Output Word Clock for Port A	GPIO35
AUDO_I2SB_WCLK	AD3	O	I2S Stereo Channel Output Word Clock for Port B	GPIO38
AUDO_MUTE	AE1	O	TTL output for mute audio via ext amplifier by GPIO	GPIO63
AUDO_SPDIF_OUT	AD1	O	SPDIF Audio output	GPIO41
AUD_OUT_HP_L	T1	AO	Audio headphone out ; dac0 OR dac2	
AUD_OUT_HP_R	T2	AO	Audio headphone out ; dac1 OR dac3 -	
AUD_OUT1_L	R1	AO	Audio analog output main channel left speaker. (DAC0, or any AUD_IN_L* or AUD_MONO_IN)	
AUD_OUT1_R	R2	AO	Audio analog output main channel left speaker. (DAC1, or any AUD_IN_R* or AUD_MONO_IN)	
AUD_OUT2_L	T3	AO	Audio analog output main channel left speaker. (DAC0 or DAC2, or any AUD_IN_L* or AUD_MONO_IN)	
AUD_OUT2_R	T4	AO	Audio analog output main channel left speaker. (DAC1 or DAC3, or any AUD_IN_R* or AUD_MONO_IN)	
LS_OUT_L	R3	AO	Audio DAC2 output (for loudspeaker left output)	
LS_OUT_R	R4	AO	Audio DAC3 output (for loudspeaker right output)	
LS_OUT_SW	R5	AO	Audio DAC4 output (for subwoofer)	

Table 11. Host I/F

Signal	Ball No	IO Type	Description	Alternate Function
HOST_A7	AJ14	O	Host Address bit 7	GPO15
HOST_A8	AE13	O	Host Address bit 8	GPO16
HOST_A9	AF12	O	Host Address bit 9	GPO17
HOST_A14	AF11	O	Host Address bit 14	GPO22
HOST_A15	AG11	O	Host Address bit 15	GPO23
HOST_A16	AH11	O	Host Address bit 16	GPO24
HOST_A17	AE10	O	Host Address bit 17	GPO25
HOST_A18	AF13	O	Host Address bit 18	GPO26
HOST_A19	AG13	O	Host Address bit 19	GPO27
HOST_A20	AE12	O	Host Address bit 20	GPIO156
HOST_A21	AD13	O	Host Address bit 21	GPIO157

Signal	Ball No	IO Type	Description	Alternate Function
HOST_A22	AH13	O	Host Address bit 22	GPIO158
HOST_A23	AJ11	O	Host Address bit 23	GPIO159
HOST_A24	AD9	O	Host Address bit 24	GPIO62
HOST_ACK	AG3	I	Host ACK	GPI7
HOST_D8	AF7	IO	Host Data bit 8	GPIO148
HOST_D9	AH8	IO	Host Data bit 9	GPIO149
HOST_D10	AE8	IO	Host Data bit 10	GPIO150
HOST_D11	AJ9	IO	Host Data bit 11	GPIO151
HOST_D12	AG9	IO	Host Data bit 12	GPIO152
HOST_D13	AE9	IO	Host Data bit 13	GPIO153
HOST_D14	AH10	IO	Host Data bit 14	GPIO154
HOST_D15	AF10	IO	Host Data bit 15	GPIO155
HOST_BOOT_CS_N	AG6	O	Host Boot Chip select (active low); only used for NOR flash	GPIO66
HOST_DEV_CS0_N	AJ4	O	Host Device Chip select 0 (active low)	GPIO65
HOST_DEV_CS1_N	AF4	O	Host Device Chip select 1 (active low)	GPIO61
HOST_DEV_CS2_N	AE5	O	Host Device Chip select 2 (active low)	GPIO60
HOST_READY	AH5	IO	Ready / Busyn flag from Nand flash; connected internally to GPIO64 to allow Mips to poll or receive irq to monitor Nand flash status.	GPIO64
POD_HOST_A0	AJ15	O	CableCard/CI Host Address bit 0	GPO8
POD_HOST_A1	AH15	O	CableCard/CI & Host Address bit 1	GPO9
POD_HOST_A2	AG15	O	CableCard/CI & Host Address bit 2	GPO10
POD_HOST_A3	AE14	O	CableCard/CI & Host Address bit 3	GPO11
POD_HOST_A10	AG12	O	CableCard/CI & Host Address bit 10	GPO18
POD_HOST_A11	AH12	O	CableCard/CI & Host Address bit 11	GPO19
POD_HOST_A12	AJ12	O	CableCard/CI & Host Address bit 12	GPO20
POD_HOST_A13	AE11	O	CableCard/CI & Host Address bit 13	GPO21
POD_HOST_D0	AG7	IO	CableCard/CI & Host Data bit 0	
POD_HOST_D1	AJ8	IO	CableCard/CI & Host Data bit 1	
POD_HOST_D2	AG8	IO	CableCard/CI & Host Data bit 2	
POD_HOST_D3	AE7	IO	Host Data bit 3	
POD_HOST_D4	AH9	IO	CableCard/CI & Host Data bit 4	
POD_HOST_D5	AF9	IO	CableCard/CI & Host Data bit 5	
POD_HOST_D6	AJ10	IO	CableCard/CI & Host Data bit 6	
POD_HOST_D7	AG10	IO	CableCard/CI & Host Data bit 7	
POD_IORD_HOST_A6	AH14	O	CableCard/CI IO Read Enable or Host Address bit 6	GPO14
POD_IOWR_HOST_A5	AG14	O	CableCard/CI IO Write Enable or Host address bit 5	GPO13
POD_OE_HOST_RD	AF8	O	CableCard/CI Output Enable or Host read enable	GPO29
POD_WE_HOST_WR	AJ13	O	CableCard/CI Write Enable or Host write enable	GPO28
PODREG_HOST_A4	AF14	O	CableCard/CI Reg or Host Address bit 4	GPO12

Table 12. CableCARD/CI Additional Signals

Signal	Ball No	IO Type	Description	Alternate Function
OOB_CRX	W2	I	Clock received from Out of band tuner	GPIO43
OOB_CTX	W1	I	Transmit clock received from Out of band tuner	GPIO42
OOB_DRX	W3	I	Data received from Out of band tuner	GPIO44
POD_A4_CTX	AE3	IO	CableCard/CI Address bit 4 or Out of Band Transmit Clock received from OOB tuner and sent to CableCard	GPIO19
POD_A5_ITX	AF3	IO	CableCard/CI Address bit 5 or ITX from CableCard	GPIO20
POD_A6_ETX	AF2	IO	CableCard/CI Address bit 6 or ETX from CableCard	GPIO21
POD_A7_QTX	AG2	IO	CableCard/CI Address bit 7 or QTX from CableCard	GPIO22
POD_A8_CRX	AF1	IO	CableCard/CI Address bit 8 or CRX to CableCard (Clock from Out Of Band Demod to CableCard)	GPIO23
POD_A9_DRX	AG1	IO	CableCard/CI Address bit 9 or DRX to CableCard	GPIO59
POD_A14_MCLKO	W5	IO	CableCard/CI Address bit 14 or MPEG Transport Input Clock (CableCard)	GPIO24
POD_BVD1_MOSTRT	Y4	I	Battery Voltage Detect pin 1 or MPEG Transport Input Start (TOP=top-of-packet)	GPIO27
POD_BVD2_MOVAL	Y5	I	Battery Voltage Detect pin 2 or MPEG Transport Input Valid	GPIO26
POD_CD_1	V5	I	CableCard/CI Card Detect 1	GPIO46
POD_CD_2	V4	I	CableCard/CI Card Detect 2	GPIO45
POD_CE_1	AE4	IO	CableCard/CI Chip Enable 1	GPIO18
POD_CE_2	W4	IO	CableCard/CI Chip Enable 2	GPIO47
POD_D8_MDO0	Y3	I	CableCard/CI Data bit 8 or MPEG data 0 from CableCard/CI	GPIO96
POD_D9_MDO1	AA3	I	CableCard/CI Data bit 9 or MPEG data 1 from CableCard/CI	GPIO97
POD_D10_MDO2	Y2	I	CableCard/CI Data bit 10 or MPEG data 2 from CableCard/CI	GPIO98
POD_D11_MDO3	AA2	I	CableCard/CI Data bit 11 or MPEG data 3 from CableCard/CI	GPIO99
POD_D12_MDO4	AB2	I	CableCard/CI Data bit 12 or MPEG data 4 from CableCard/CI	GPIO100
POD_D13_MDO5	Y1	I	CableCard/CI Data bit 13 or MPEG data 5 from CableCard/CI	GPIO101
POD_D14_MDO6	AA1	I	CableCard/CI Data bit 14 or MPEG data 6 from CableCard/CI	GPIO102
POD_D15_MDO7	AB1	I	CableCard/CI Data bit 15 or MPEG data 7 from CableCard/CI	GPIO103
POD_DETECT_N	AD4		Tri-state control of CableCard/CI interface	GPIO16
POD_DIR_N	AD5	O	External pod buffer control;'0'- write; '1'- read.	GPIO17
POD_READY_IRQ_N	V2	IO	CableCard/CI Ready or Interrupt Request (after personality change in CA/CableCard/CI)	GPIO49

Signal	Ball No	IO Type	Description	Alternate Function
POD_RESET	V1	IO	CableCard/CI Reset (Active High)	GPIO50
POD_VS2_MCLKO	W6	I	Voltage Sense pin-2 or MPEG Transport Input Clock (CI)	GPIO25
POD_WAIT_N	V3	I	CableCard/CI Wait	GPIO48

Table 13. HDMI

Signal	Ball No	IO Type	Description	Alternate Function
ARX0M	A9	I	HDMI input pair 0	
ARX0P	B9	I	HDMI input pair 0	
ARX1M	A8	I	HDMI input pair 1	
ARX1P	B8	I	HDMI input pair 1	
ARX2M	A7	I	HDMI input pair 2	
ARX2P	B7	I	HDMI input pair 2	
ARXCM	A10	I	HDMI clock input pair	
ARXCP	B10	I	HDMI clock input pair	
BRX0M	A5	I	HDMI input pair 0	
BRX0P	B5	I	HDMI input pair 0	
BRX1M	A4	I	HDMI input pair 1	
BRX1P	B4	I	HDMI input pair 1	
BRX2M	A3	I	HDMI input pair 2	
BRX2P	B3	I	HDMI input pair 2	
BRXCM	A6	I	HDMI clock input pair	
BRXCP	B6	I	HDMI clock input pair	
HDMI_A_HPD	B11		HDMI hot plug detect – for indicating to HDMI source when ready to receive	GPIO9
HDMI_B_HPD	C11		HDMI hot plug detect – for indicating to HDMI source when ready to receive	GPIO67
HDMI_CEC	A11		HDMI CEC interface	GPIO10
REXT	D9	AI	External termination resistor.	

Table 14. Two-Wire

Signal	Ball No	IO Type	Description	Alternate Function
2WIRE_M0_SCL	E20	O	2-wire master clock output	
2WIRE_M0_SDA	F20	IO	2-wire master data IO	
2WIRE_M1_SCL	D20	O	2-wire master clock output	GPIO52/UART2_RXD
2WIRE_M1_SDA	D21	IO	2-wire master data IO	GPIO51/UART2_TXD
2WIRE_S0_SCL	A12	I	2-wire slave clock IO (for VGA slave)	GPIO54
2WIRE_S0_SDA	B12	IO	2-wire slave data IO (for VGA slave)	GPIO53
2WIRE_S1_SCL	C12	IO	2-wire slave clock input for HDMI connector 'A' HDCP and DDC inputs. Also can be master clock output for programming external EDID.	GPIO56
2WIRE_S1_SDA	D12	IO	2-wire slave data IO for HDMI connector 'A' HDCP, DDC input, and master output for reading or programming external EDID	GPIO55
2WIRE_S2_SCL	D11	IO	2-wire slave clock input for HDMI connector 'A' HDCP and DDC inputs. Also can be master clock output for	GPIO58

Signal	Ball No	IO Type	Description	Alternate Function
			programming external EDID.	
2WIRE_S2_SDA	E11	IO	2-wire slave data IO for HDMI connector 'B' HDCP, DDC input, and master output for programming external EDID	GPIO57

Table 15. System Interface

Signal	Ball No	IO Type	Description	Alternate Function
DFSYNC	E13	IO	DFSYNC_IN, DFSYNC_OUT	GPIO8
PBIAS	B21	O	Panel bias enable output	GPO6
PPWR	A21	O	Panel power enable output	GPO7
PWM0	D13	IO	PWM0 output; also used as GPIO4	GPIO4
PWM1	C13	IO	PWM1 output; also used as GPIO5	GPIO5
PWM2	B13	IO	PWM2 output; also used as GPIO6	GPIO6
PWM3	A13	IO	PWM3 output; also used as GPIO7	GPIO7
RESET_N	D23	I	Hardware Reset input (active low)	

Table 16. LBADC

Signal	Ball No	IO Type	Description	Alternate Function
LBADC_33	D24	AP	3.3V LBADC Analog voltage supply	
LBADC_GND	E23	AG	Analog 3.3v ground	
LBADC_IN1	C25	AI	Analog input to low bw adc	GPO0
LBADC_IN2	C24	AI	Analog input to low bw adc	GPO1
LBADC_IN3	B25	AI	Analog input to low bw adc	GPO2
LBADC_IN4	B24	AI	Analog input to low bw adc	GPO3
LBADC_IN5	A25	AI	Analog input to low bw adc	GPO4
LBADC_IN6	A24	AI	Analog input to low bw adc	GPO5
LBADC_RETURN	F22	AI	Common analog input signal return (can share pin with analog 3.3V ground).	

Table 17. UART

Signal	Ball No	IO Type	Description	Alternate Function
UART1_CTS	C19	O	UART 1, clear to send	GPIO0
UART1_RTS	C20	O	UART 1, request to send	GPIO3
UART0_RXD	A19	I	UART 0, receive data	GPI0
UART1_RXD	A20	I	UART 1, receive data	GPIO1
UART0_TXD	B19	O	UART 0, transmit data	GPI1
UART1_TXD	B20	O	UART 1, transmit data	GPIO2
UART2_RXD	D20	I	UART 2, receive data	GPIO52/2WIRE_M1_SCL
UART2_TXD	D21	O	UART 2, transmit data	GPIO51/2WIRE_M1_SDA

Table 18. E/JTAG

Signal	Ball No	IO Type	Description	Alternate Function
EJ_DINT	B29	I/O	EJTAG interrupt	
EJ_RST_N	A29	I	EJTAG Reset input	
TCK	B28	I	JTAG test clock	
TDI	B27	I	JTAG test data in	
TDO	A27	O	JTAG test data out. Tri-State output	
TMS	A28	I	JTAG test mode select	
TRST	B26	I	JTAG test reset	

Table 19. USB

Signal	Ball No	IO Type	Description	Alternate Function
USB_FLAG	C29	I	Input from of chip regulator, which sets to one when over current condition exists USB	GPI11
USB_PWREN	C28	O	USB host controller to enable power to USB connector	GPO30
USBPHY_PADM	AH20	IO	D- line of USB2.0 transceiver	
USBPHY_PADP	AJ20	IO	D+ line of USB2.0 transceiver	
USBPHY_VRES	AG20		Connect to external 6.2K resistor (the 6.195K resistor from VRES pin to PCB ground. Put a 0.1 Uf decoupling capacitor in parallel with the resistor.	

Table 20. IR

Signal	Ball No	IO Type	Description	Alternate Function
IRDATA	C21	I	Infra-red decoder input	GPI8

Table 21. Test Modes

Signal	Ball No	IO Type	Description	Alternate Function
TESTMODE0	C27	I	This pin (TESTMODE0) is lsb together with TESTMODE1 for two bit word to determine chip test/functional mode as follows: 00 = normal mission mode (system without external reset controller); POR circuit is enabled (100 msec reset); 01 = Mission mode with POR circuit bypassed to allow external reset controller (also used for ATE functional vectors testing); 10 = Boundary Scan with POR bypass (Board level boundary scan testing and ATE parametric testing; 11 = Scan mode without POR bypass (used for ATE scan test and POR test).	
TESTMODE1	C26	I	Test mode sel 1 – see TESTMODE0 description	

Table 22. Not Connected

Signal	Ball No	IO Type	Description	Alternate Function
NC	A26 F[7..11]		Do not connect	

Table 23. Power and Ground

Signal	Ball No	IO Type	Description	Alternate Function
ADC_GNDA	E3 F1 F5 G3 G5 H5 J5 K5	AG		
ADC_GND12	D4 E5 F6	AG		
ADC_VDD12	A1 B2 C3	AP	1.2V ADC digital voltage supply	
ADC_VDDA33	E4 F4 G4 H4 K4	AP	3.3V ADC analog voltage supply	
AUD_AVDD12	P6 R6		1.2V Audio digital voltage supply	
AUD_AVSS12	M6 N6	AG	GND_AUD_12	
AUD_AVDD33	T6		3.3V Audio voltage supply	
AUD_AVDD33	T5		VCCGR1_AUD_33	
AUD_AVDD33	U5		VCCGR2_AUD_33	
AUD_AVDD33	U4		VCC5_AUD_33	
AUD_AVDD33	U3		VCC4_AUD_33	
AUD_AVSS33	P1	AG	GND3_AUD_33	
AUD_AVSS33	P2	AG	GND1_AUD_33	
AUD_AVSS33	P3	AG	GNDGR2_AUD_33	
AUD_AVSS33	P4	AG	GND5_AUD_33	
AUD_AVSS33	P5	AG	GND4_AUD_33	
AUD_VCM	L5		Analog common mode voltage	
AUD_VREFN	L4		Analog reference voltage negative	
AUD_VREFP	L3		Analog reference voltage positive	
AUD_HP_AVDD33	U2		3.3V Audio Head Phone voltage supply	
AUD_HP_AVSS33	U1	AG	GNDHP_AUD_33	
CVDD12	L[11..12] L[17..19]	P	1.2V Core digital supply	

Signal	Ball No	IO Type	Description	Alternate Function
	M[11..12] M[17..19] V[11..13] V[18..19] W[11..13] W[18..19]			
CVSS	D26 D28 E24 E26 E28 G24 G26 G27 J24 J26 J28 L24 L26 L28 N27 N28 P26 T24 T26 V29 W24 W26 W28 AA24 AA26 AA27 AC24 AC26 AC28 AE24 AE26 AE28	IO G	DDR IO VSS - connect to common ground.	
CVSS	L[13..16] M[13..16] N[11..19] P[11..19] R[11..19] T[11..19] U[11..19] V[14..17] W[14..17]	IO G	Core VSS connect to common ground	

Signal	Ball No	IO Type	Description	Alternate Function
DDR_VDD	D25 D27 D29 F25 F27 F29 H24 H27 H29 K25 K27 K29 M25 M29 N24 P28 R25 U25 V27 Y25 Y27 Y29 AB24 AB27 AB29 AD25 AD27 AD29 AF25 AF27 AF29		1.8V DDR2 IO voltage supply	
DDR_VDDI	F23 AD23	I	3.3V DDR input Rx VDD voltage supply, connect to PCB 3.3V	
DDR_VRF_0	G25	I	DDR voltage reference input DDR2 = 0.9V	
DDR_VRF_1	R24	I	DDR voltage reference input DDR2 = 0.9V	
DDR_VRF_2	AA25	I	DDR voltage reference input DDR2 = 0.9V	
DDRPLL_AGND	E21	AG	Short to DDRPLL_AGND33, share ball E21	
DDRPLL_AVDD12	D22	AP	1.2V DDR Analog PLL voltage supply	
DDRPLL_AVDD33	E22		3.3V Analog DDR clock PLL voltage supply	
DLL_VAA0	M28		3.3V DLL VAA0 voltage supply, connect to 3.3V	
DLL_VAA1	V24		3.3V DLL VAA1 voltage supply, connect to 3.3V	
HDMI_GNDA	A2	G		

Signal	Ball No	IO Type	Description	Alternate Function
	C[5..10] D[7..8]			
HDMI_VDDA33	C4 D[5..6] E[6..7]	P	3.3V HDMI/DVI voltage supply	
HDMI_VDD12	D10 E[8..10]	P	1.2V HDMI/DVI digital voltage supply	
IOVDD33	F[12..16] F[18..19] AD8 AD[10..12] AD14 AD[16..18] U6 V6 Y6 AA6 AB6	IO P	3.3V IO voltage supply	
LVTX_PLL_VDD33	AG26	AP	3.3V Analog LVDS clock PLL voltage supply	
LVTX_VDD33	AE[21..22] AF24	AP	3.3V LVDS Tx VDD voltage supply	
LVTX_VSS	AD[21..22] AE23 AG27	AP		
RPLL_AGND	B22	AG	Analog PLL ground corresponding to 1.2V	
RPLL_AGND	C22	AG	Analog PLL ground corresponding to 3.3V	
RPLL_AVDD12	C23	AP	1.2V Analog PLL voltage supply	
RPLL_AVDD33	B23	AP	3.3V Analog Ref clock PLL voltage supply	
USB_GND	AD20 AJ19	AG	Analog ground of UTMI+ PHY, connect to PCB ground.	
USB_GND	AF19	G	Digital ground of UTMI+ PHY, connect to PCB ground.	
USB_AVDD12	AE20	P	1.2V USB Core voltage supply	
USB_AVDD33	AF20	P	3.3V USB digital voltage supply, connect to PCB 3.3 volt power	
USB_AVDD33	AG19 AH19	AP	Analog 3.3 volt power pad, connect to PCB 3.3 volt power	
VDAC_AVSS33	AD7	AP	DAC Analog VSS corresponding to 3.3V	
VDAC_AVSS33	AG4	AP	DAC Analog VSS for band gap	
VDAC_AVDD33	AE6 AF[5..6]	AP	3.3V Video DAC Analog voltage supply	
VDAC_VDD12	AD6	AP	1.2V Video DAC digital voltage supply	
VDAC_VSS12	AC6	AP	DAC Core VSS corresponding to 1.2V	

4 RESET

The reset module generates all the resets required for FLI106xx. Functionality includes:

- Reset for all the on-chip peripherals.
- Separate reset signal for MIPS and DMA.
- Reset for all external devices.

4.1 RESET OPTIONS

The chip supports the following reset options:

- Hard Reset from Power-On Reset (POR) Generator
- Hard Reset from EJTAG Probe
- Soft Reset from MIPS EJTAG Interface
- System Soft Reset from Reset Generator Registers

All of these reset sources except the last source are common for all blocks. Register-controlled block soft reset is controlled independently for each block. All configuration registers are reset to their Power on Default values on soft and hard reset.

4.1.1 POWER-ON RESET (POR)

The POR block detects the I/O supply voltage level and holds the output reset active low when the voltage drops below 2.4V. The output reset goes high after a period of t_{OR} when the voltage reaches above 2.4V. By doing this, all the internal blocks are held inactive until the supply voltage is stable.

The POR generator works as follows:

1. On power-on the block detects when the power supply reaches 2.4V and holds the reset active for a period of t_{OR}.
2. When the reset is pulled low externally using a push-button switch, from the moment the reset is released, the reset to the internal blocks and external devices is held low for t_{OR} before it can go high.
3. When the voltage supply is not clean, all the peripherals are inactive. Therefore, when the supply voltage falls below 2.4V, the output reset is held low. From the moment the power supply rises beyond 2.4V, the output reset is held low for t_{OR} before it becomes high.

Table 24. Voltage and Timing Parameters for POR Module

Parameter	Min	Typ	Max
Output Reset Time t_{OR}	120ms	140ms	160ms
Vdd Slew Rate	0	-	-
Push Button ON Time	200us	-	-
Vdd	3.0v	3.3v	3.6v
Min Vdd for Operation	2.6v	2.6	2.6
Power-ON Threshold	2.4v	2.4	2.4
Power-DOWN Threshold	2.4v	2.4	2.4

Table 25. POR Table of Operation

Vdd Condition (3.3)	Remarks
0 <Vdd <2.4	Power-Down, the output reset is held low as long as this condition persists
Vdd ≥ 2.4	Power-On, the output reset goes high after t_{OR} from the moment Vdd crosses 2.4. The Vdd should be >2.4 for t_{OR}; if not the reset should be maintained low
When Vdd drops below 2.4	Power-Down again, the output reset is held low as long as this condition prevails. Output reset goes high after t_{OR} from the point the Vdd again crosses 2.4v and stays above 2.4v for around t_{OR}
Push button reset time < 200us	The reset is not generated, goes back high the moment it is released
Push button time > 200us	The reset is held low for t_{OR} from the time the reset is released, and then goes high. Vdd should be >2.4 during this period

Two features for system robustness are integrated into the device:

- A low-voltage monitor function
- A reset button de-glitch monitor function

These integrated features remove the cost associated with system power and reset button monitoring that are normally external features.

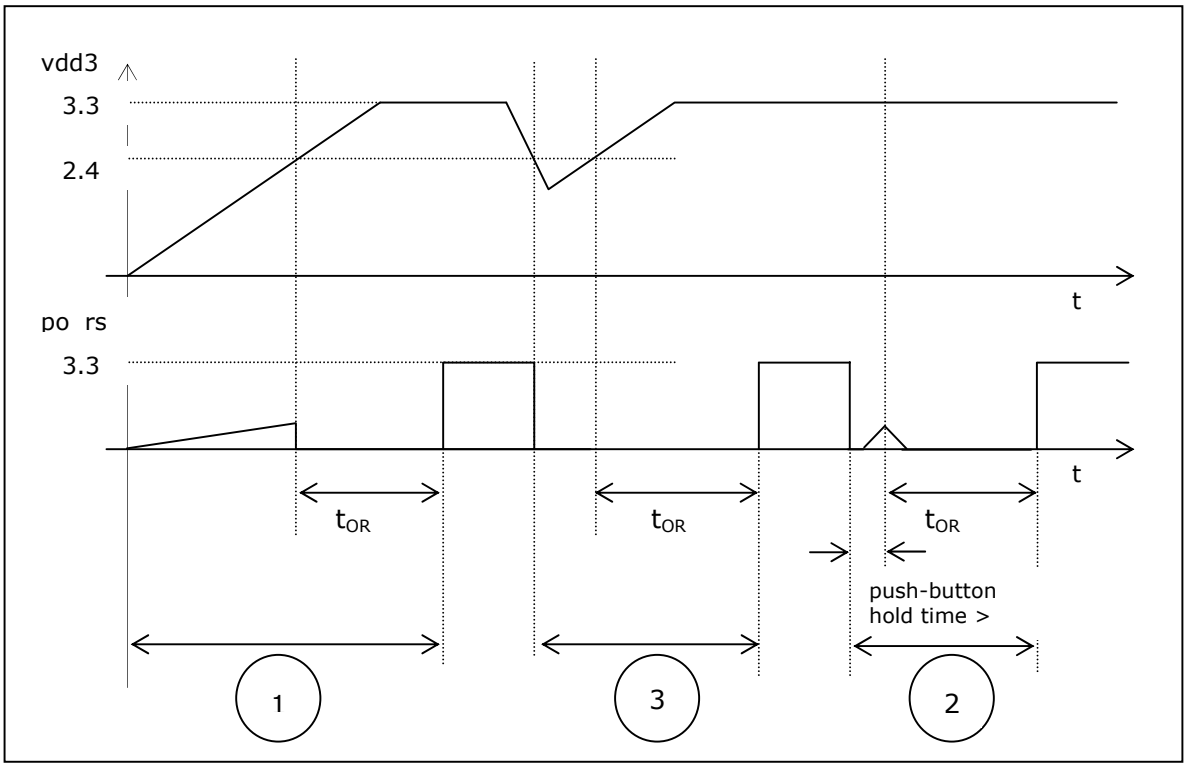


Figure 8. POR Modes of Operation

4.1.2 EJTAG HARD RESET

The reset for EJTAG probe is a hard reset for the entire chip. This reset should be held active low by the probes for a minimum period of 100 ms, and is configurable for up to 1 sec.

4.1.3 EJTAG SOFT RESET

The EJTAG based debugger can either soft-reset the entire chip or reset all internal blocks except MIPS by programming the MIPS EJTAG control register.

4.1.4 SYSTEM SOFT RESET

Setting bit SYS_RESET of BLK_RESET_CTRL register to '1' causes Soft Reset of all FLI106xx blocks, including MIPS 4KEc. This bit is cleared by the reset. This does not affect other chips in the system.

4.1.5 SYSTEM RESET ARISING OUT OF REPEATED WATCHDOG TIMEOUTS

The reset generator implements a watchdog counter, counting watchdog assertions. Each time the watchdog fires (generating MIPS NMI), the watchdog counter is incremented. When the counter reaches a pre-programmed threshold, it generates a system soft reset. This reset can be enabled/disabled by programming the watchdog registers in the chip.

4.2 POWER SEQUENCING

The chip requires three power supply inputs for normal operation – 3.3V (I/O), 1.8V (DDR), and 1.2V (core). At any time during the power-up sequence (see Figure 9), the actual voltage of the 3.3V IOVDD power supply should always be equal to or higher than the actual voltage of the 1.8V and 1.2V Core VDD power supply. It is recommended that the power supply arrives in the following order:

1. 3.3V (IO)
2. 1.2V
3. 1.8V (DDR2)

The maximum delay between the supplies cannot be determined because the delay depends on the slew rate of the ramping up operation. To save the transient current during power down, it is advised that the core voltage (1.2V) be powered down first and then the DDR VDD (1.8V) and the IO voltage (3.3V).

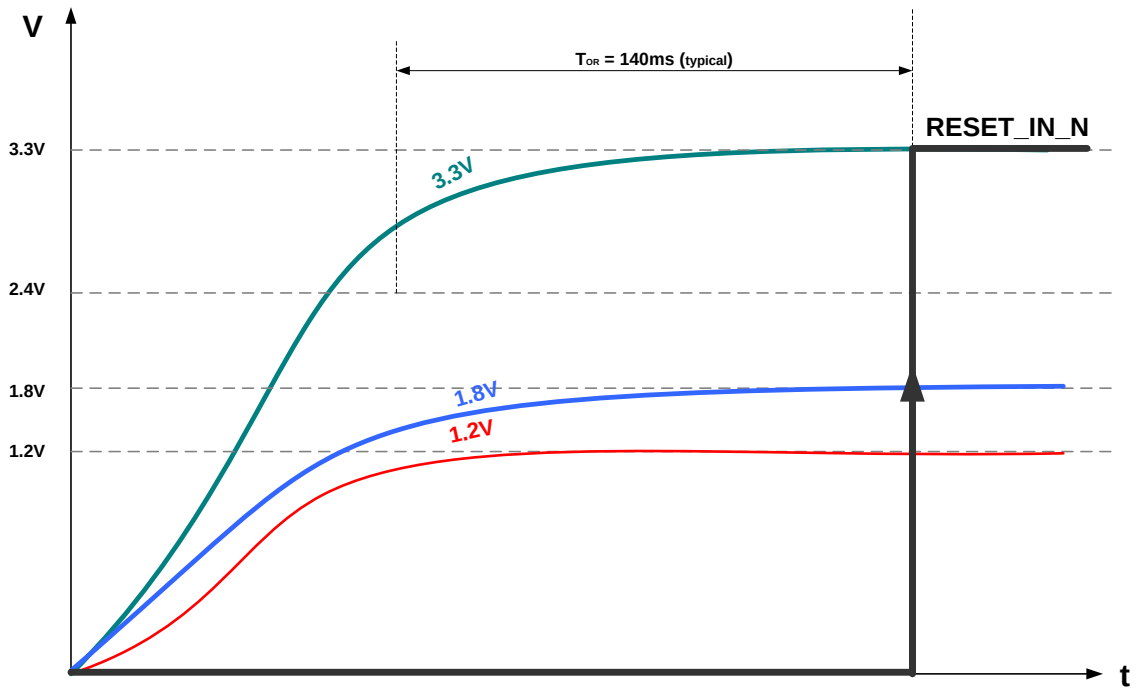


Figure 9. Power-Up Sequencing Operation

Additionally, the 1.2V core VDD along with 1.8V DDR2 and 3.3V supply must be active for at least 20ms before the rising edge of the chip RESET_IN_N signal during the chip power-up sequence. The rising edge of RESET_IN_N signal is used to latch the bootstrap configurations, so its correct timing relationship to the core VDD is critical for the proper functioning of the chip.

5 BOOT PROCESS

5.1 IROM

FLI106xx implements 2Kx32-bit Instruction ROM. IROM is mapped to the physical address range 1FC0_0000 – 1FC0_1FFF (MIPS boot address is 1FC0_0000). The same address range is shadowed in NOR FLASH, if connected on host interface. IROM may be bypassed to boot directly from NOR FLASH, controlled by a bootstrap.

IROM is only required if:

4. Secure Boot, as part of OpenCable standards, is required
5. Support for NAND FLASH is required

If these are not required, IROM can be bypassed for NOR FLASH.

Clearing CRO IROM_ENABLE register permanently remaps IROM address space to host interface (identical effect to bypass bootstrap). This register is cleared by the last instruction in IROM. IROM then becomes inaccessible until Reset is asserted.

IROM is always accessed 32-bit at a time; no burst or byte access to IROM is supported.

5.2 SECURE BOOT

The secure boot process includes an 8KB IROM that is accessible to MIPS. After the boot process completes, the IROM becomes inaccessible.

Note: that a bootstrap option is available for IROM bypass to boot from NOR FLASH.

5.3 BOOT SEQUENCE

The boot sequence of the device is a multiple-stage procedure that utilizes internal IROM code and External bootstrap configuration. The system boots from the internal ROM. Therefore, the boot address of the MIPS processor must be mapped to the internal ROM.

Note that booting from the IROM can be bypassed by using a boot strapping option. After the system boots up from the internal ROM, it checks for bootstrap pins. These pins help to figure out the types of FLASH that are interfaced to the chip and external ROM bootup sector.

Boot Option	HOST_A3	HOST_A4	HOST_A5	HOST_A6
16 bit NAND – Large Page	X	1	1	1
16 bit NAND – Small page	X	1	0	1
16 bit NOR	1	0	X	X
8 bit NAND – Large Page	X	1	1	0
8 bit NAND – Small Page	X	1	0	0
8 Bit NOR	0	0	X	X

The IROM then begins signature authentication of the code in the external FLASH and enables the secure boot process if it finds a valid signature. After the IROM code has successfully read and decrypted the entire FLASH content into the DDR, the control jumps to the external ROM boot area (at this point, IROM is no longer accessible by MIPS and an exception is generated if the IROM address space is accessed). This second stage boot loader will have sufficient code to initialize the remainder of the chip.

The next stage of the boot process loads the operating system and user-defined applications.

6 BOOTSTRAP CONFIGURATION PINS

During hardware reset, the external ROM address pins POD_HOST_A*x* and POD_*xx*_HOST_A*x* are configured as inputs. On the negating edge of RESETn, the value on these pins is latched and stored. This value is readable by the On-Chip Microcontroller (OCM) to provide system configuration information. This process is called "boot-strapping".

Table 26. Bootstrap Signals

Bootstrap	Pin Name	No	Description
BSTRAP_BOOT_MODE	POD_HOST_A0 POD_HOST_A1	AJ15 AH15	Pins POD_HOST_A[1:0] indicate on chip hardware the host interface configuration to use after hard reset: A1;A0 = 00 = Function test, vendor mode. A1;A0 = 01 = Function test, vendor mode. A1;A0 = 10 = Boot from FLASH A1;A0 = 11 = Boot from IROM
BSTRAP_EXT_OSC	POD_HOST_A2	AG15	Pin POD_HOST_A2 indicates: 0 = Internal osc 1 = External osc
BSTRAP_16BIT_FLASH	POD_HOST_A3	AE14	Pin POD_HOST_A3 indicates type of memory for external boot FLASH. 0 = 8-bit FLASH 1 = 16-bit FLASH
BSTRAP_NAND_FLASH_EN	PODREG_HOST_A4	AF14	Pin PODREG_HOST_A4 indicates type of memory for external boot FLASH. 0 = NOR FLASH 1 = NAND FLASH
BSTRAP_PAGESIZE	POD_IOWR_HOST_A5	AG14	Pin POD_IOWR_HOST_A5 indicates page size for off chip NAND FLASH. 0 = Small page NAND FLASH 1 = Large page NAND FLASH
BSTRAP_NAND_FLASH_DWIDTH	POD_IORD_HOST_A6	AH14	Pin POD_IORD_HOST_A6 indicates data width for NAND FLASH (used by IROM boot s/w only). 0 = 8-bit NAND FLASH 1 = 16-bit NAND FLASH
BSTRAP_NOR_FLASH_SEL	HOST_A7	AJ14	Pin HOST_A[7] selects whether parallel NOR flash or SPI flash is used for boot when BSTRAP_BOOT_MODE=10 (IROM bypass). Ignored if BSTRAP_BOOT_MODE != 10. 0 = boot from parallel NOR flash 1 = boot from SPI flash
BTSTRAP_SPI_TYPE_SEL	HOST_A18	AF13	Pin HOST_A[18] indicates whether a standard or Atmel type SPI NOR flash is used for boot. This bit is only meaningful if BSTRAP_BOOT_MODE=10 and BSTRAP_NOR_FLASH_SEL=1 0 – standard SPI external memory interface; 1 – ATMEL data flash external memory interface.

7 CLOCK GENERATION

The FLI106xx features the following clock inputs. All additional clocks are internal clocks derived from one or more of these:

Table 27. List of Primary Input Clocks

Clock Name	Frequency Range (MHz)	Description
TCLK	Typical: 19.6608 Mhz Max: 25 Mhz Min: 10 Mhz	Low jitter clock input from external oscillator using pad or external crystal connected to on chip oscillator. Used as a reference for PLL(s).
TS_IN_CLK	<= 81Mhz Max: tbd Mhz Min: tbd Mhz	Clock for transport stream 0. Sourced from IC pad.
AUDO_REFCLK0	< 30MHz Max: tbd Mhz Min: tbd Mhz	Oversampling clock for use by audio DAC. Input from pad.
AUDO_REFCLK1	< 30MHz Max: tbd Mhz Min: tbd Mhz	Oversampling clock for use by audio DAC. Input from pad.
VXI_CLK	13.5 to 148.5 Max: tbd Mhz Min: tbd Mhz	Input clock from VXI video input port.
EXT_CLK[17:0]	54 Mhz Max: tbd Mhz Min: tbd Mhz	External Clock inputs from pins. Used for scan and debug.

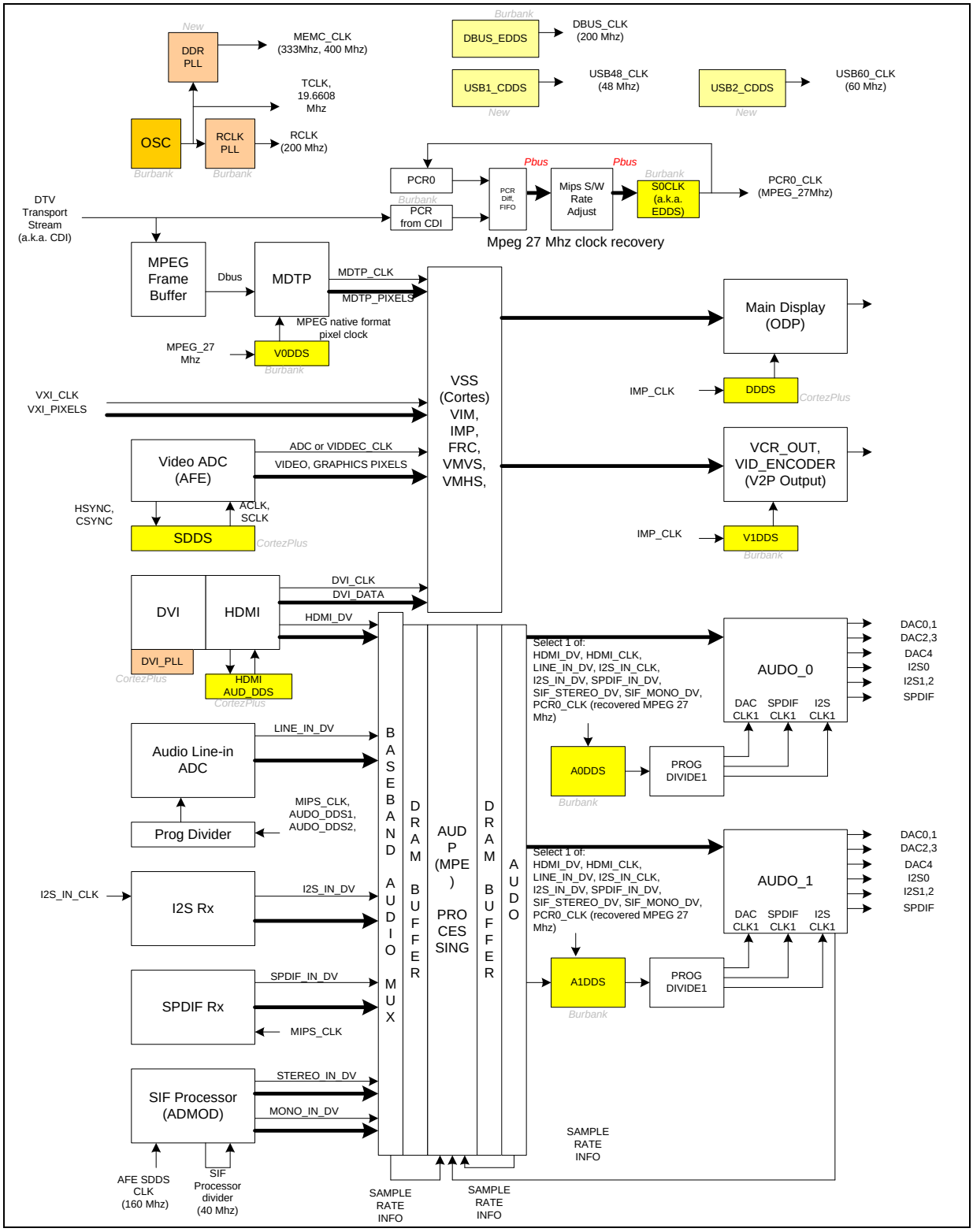


Figure 10. Clock Distribution

The TCLK oscillator circuitry supports the use of an external oscillator or a crystal resonator to generate a reference frequency source for the device.

7.1 USING THE INTERNAL OSCILLATOR WITH AN EXTERNAL CRYSTAL

The option for providing a clock reference is to use the internal oscillator with an external crystal. The oscillator circuit is designed to provide a very low jitter and very low harmonic clock to the internal circuitry of the device. An Automatic Gain Control (AGC) is used to ensure startup and operation over a wide range of conditions. The oscillator circuit also minimizes the overdrive of the crystal, which reduces its aging.

When the FLI106xx is in reset mode, the state of the POD_HOST_A2 ball is sampled. If the POD_HOST_A2 ball is pulled low to ground with an external resistor (10KΩ recommended, 15KΩ maximum), then the internal oscillator is enabled.

In this mode, a crystal resonator is connected between TCLK and XTAL with the appropriately sized loading capacitors C_{L1} and C_{L2}. The size of C_{L1} and C_{L2} are determined from the crystal manufacturer's specification and by compensating for the parasitic capacitance of the device and the printed circuit board traces. The loading capacitors are terminated to the analog VDD power supply. This connection increases the power supply rejection ratio when compared to terminating the loading capacitors to ground.

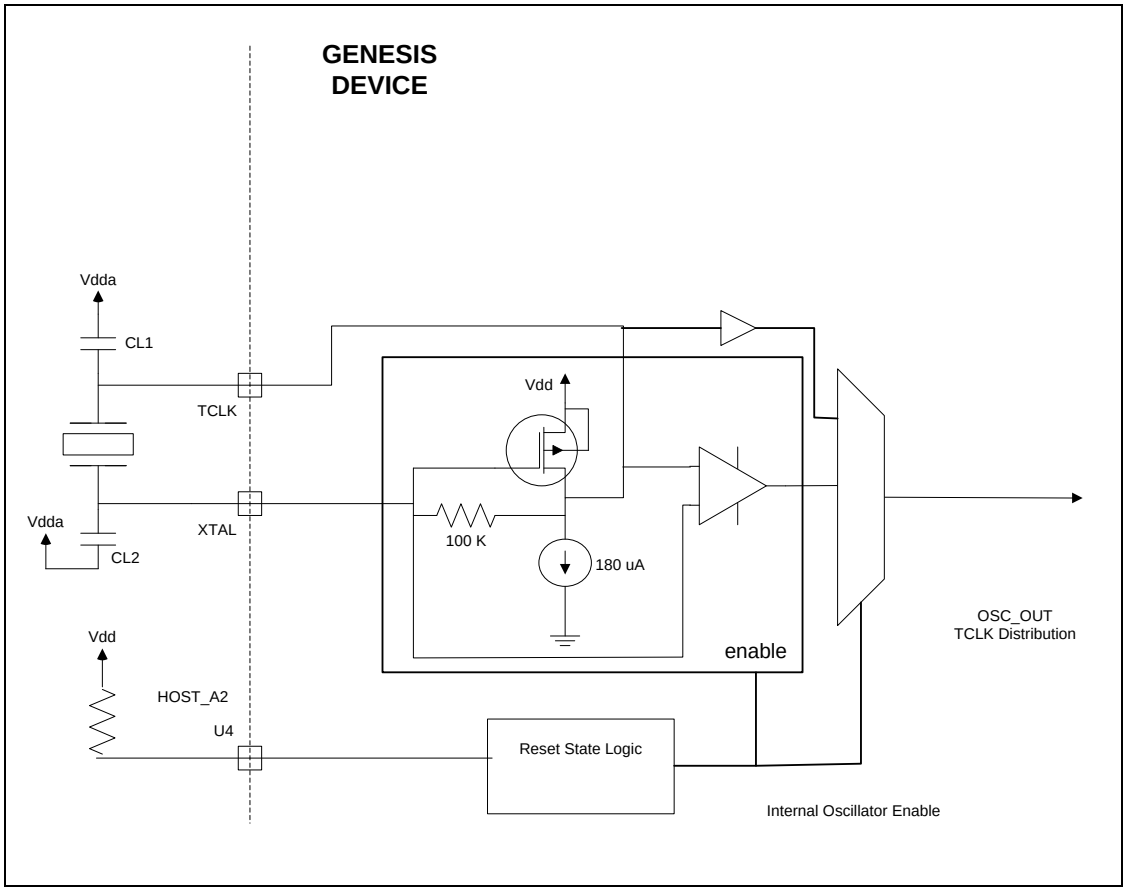


Figure 11. Using the Internal Oscillator with External Crystal

The TCLK oscillator uses a Pierce Oscillator circuit. The output of the oscillator circuit, measured at the TCLK pin, is an approximate sine wave with a bias of about 2V above ground (see Figure 12). The peak-to-peak voltage of the output can range from 250mV to 1000mV depending on the specific characteristics of the crystal and variation in the oscillator characteristics. The output of the oscillator is connected to a comparator that converts the sine wave to a square wave. The comparator requires a minimum signal level of about 50mV peak-to-peak to function correctly. The output of the comparator is buffered and then distributed to the FLI106xx circuits.

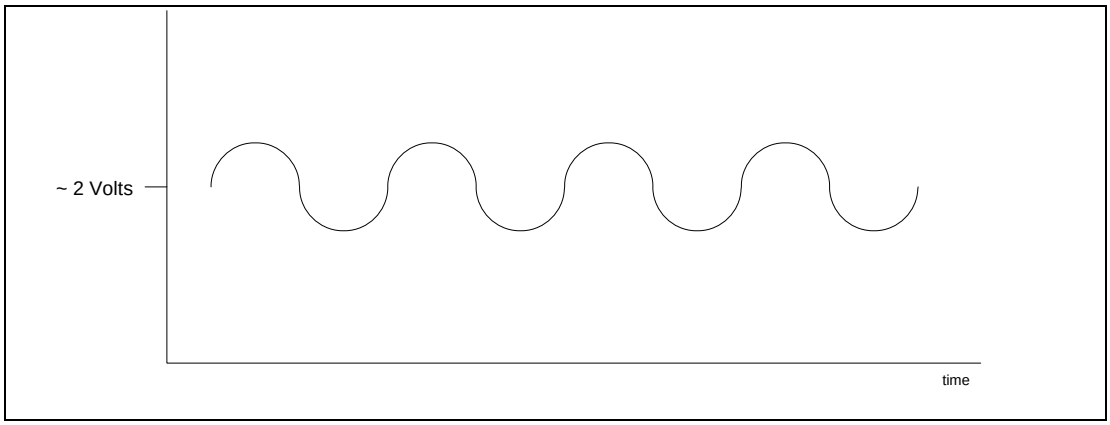


Figure 12. Internal Oscillator Output

The value of the loading capacitors used with the crystal as shown in Figure 14 is an important design parameter.

The loading capacitance (C_{load}) on the crystal is the combination of C_{L1} and C_{L2} and is calculated by:

$C_{load} = ((C_{L1} * C_{L2})/(C_{L1} + C_{L2})) + C_{shunt}$.

The shunt capacitance C_{shunt} is the effective capacitance between the XTAL and TCLK pins. For the FLI106xx, the shunt capacitance is approximately 9 pF. C_{L1} and C_{L2} are a parallel combination of the external loading capacitors (C_{ex}), the PCB board capacitance (C_{pcb}), the pin capacitance (C_{pin}), the pad capacitance (C_{pad}), and the ESD protection capacitance (C_{esd}).

The capacitances are symmetrical so that:

$C_{L1} = C_{L2} = C_{ex} + C_{PCB} + C_{pin} + C_{pad} + C_{ESD}$.

The correct value of C_{ex} must be calculated based on the values of the load capacitances.

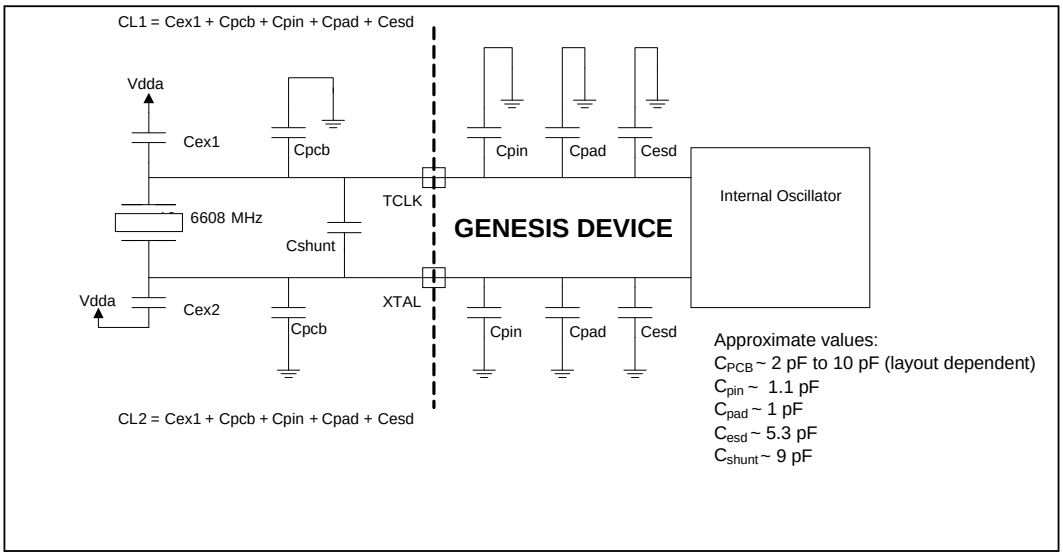


Figure 13. Sources of Parasitic Capacitance

When the oscillator circuit is used with a crystal resonator, the PCB traces should be as short as possible. The C_{load} value should not exceed the manufacturer's specification to avoid any start-up problems with the oscillator. Additionally, the crystal should be a parallel resonate-cut and the value of the equivalent series resistance must be less than 90.

It is recommended to utilize a minimum 25 ppm crystal oscillator for all applications. This will ensure optimum performance.

7.2 USING AN EXTERNAL CLOCK OSCILLATOR

Another option for providing the reference clock is to use a single-ended external clock oscillator. When the device is in reset mode, the state of the POD_HOST_A2 (ball AG15) is sampled. If POD_HOST_A2 is pulled high, then the internal oscillator is disabled and the external oscillator mode is enabled.

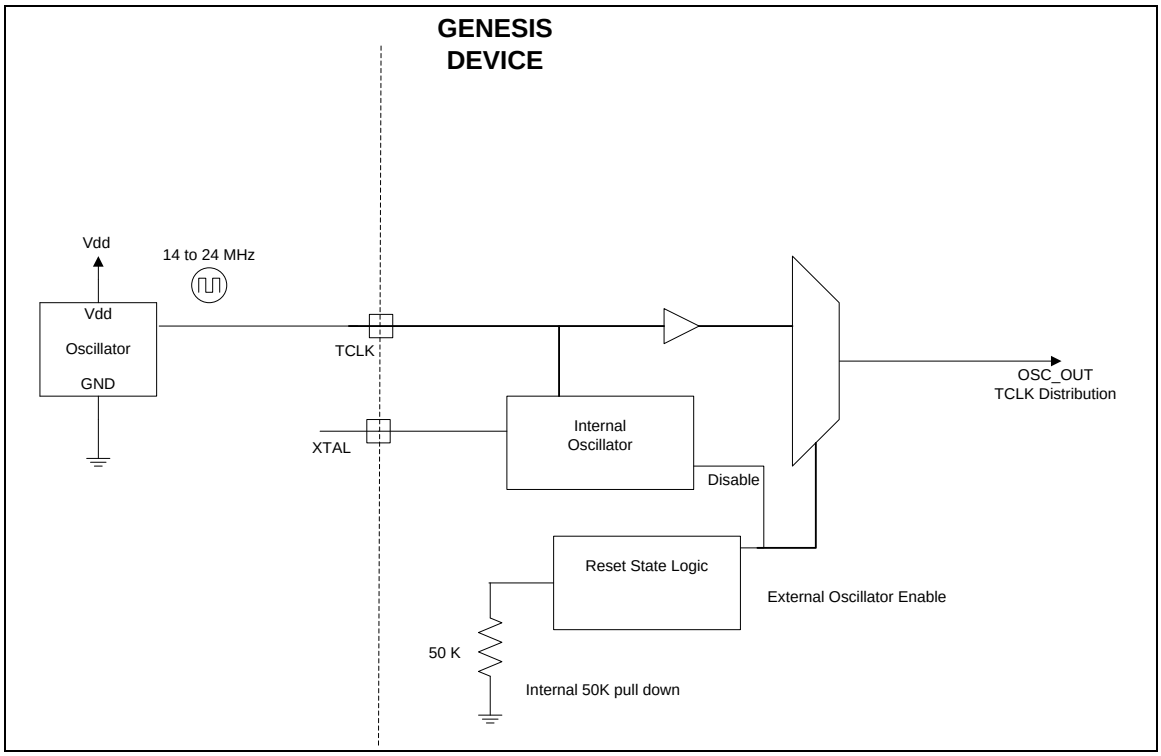


Figure 14. Using an External Single-Ended Clock Oscillator

Table 28. TCLK Specifications

Frequency	19.6608 MHz
Jitter Tolerance	+/-25 ppm
Rise Time (10% to 90%)	10 ns (typical)
Maximum Duty Cycle	48-52%

8 COMPRESSED DATA INPUT

The Compressed Data Input (CDI) interface accepts Transport Stream (TS) in serial mode and routes the data to the de-multiplexer (MPE1). The FLI106xx chip has one CDI port (CDI0) for processing transport streams coming over the RF channel.

CDI is designed to interface with a wide variety of devices via a flexible signal configuration. The CDI port supports the following input signals:

- Data (serial) (CDI0_D0 pin)
- Clock (CDI0_CLK pin)
- Sync or Top of Packet (TOP) indicator (CDI0_SYNC pin)
- Data Valid (CDI0_VALID pin)
- Error (CDI0_ERROR pin)

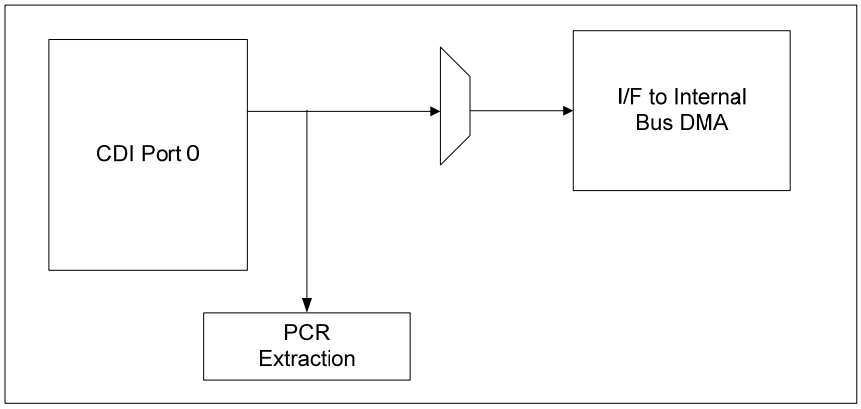


Figure 15. CDI Block Diagram

The CDI block contains an IO Control block and a Program Clock Reference (PCR) Extraction block, as shown in the figure above. CDI operates on packet boundaries and looks for the TOP/SYNC indicator to identify the start of a packet or sector. When data arrives at the CDI port, it organizes the data stream in 188 byte size packets.

In the TS mode, 4 bytes of status data is appended by CDI to each 188 byte packet.

CDI can report the following types of errors:

- **Early Top Error (ETE):** An ETE indicates that the TOP corresponding to the current buffer was an unexpected TOP.
- **Packet Length Error (PLE):** A PLE indicates that the length of the current packet exceeds the programmed length (counting from TOP).

The CDI port supports continuous, as well as gapped clock with a frequency of up to 81 MHz in serial mode. In serial mode, CDI can be programmed to receive data as Most-Significant Bit (MSB) first or Least-Significant Bit (LSB) first. The active edge can be rising edge or falling edge of the clock. If the clock is continuous, the captured data is qualified with Data Valid. Figure 16 and Figure 17 illustrate the external interface with timing diagrams.

Clock recovery can be done under software control using the PCR extraction function provided in CDI. PCR fields extracted from the incoming TS are pushed into a FIFO that can be read by software. The value of PCR timestamps and system timer clock (STC) are used to gauge the drift between local and remote transmitter clocks. Based on this drift, the software decides the amount and direction of adjustment (speed-up/slow-down) of the local reference clock. This change can be achieved on the on-chip reference PLL. An optional off-chip VCXO can be controlled using separate PWMs provided for the purpose.

8.1 CDI INTERFACE

CDI interfaces with external devices in the following ways:

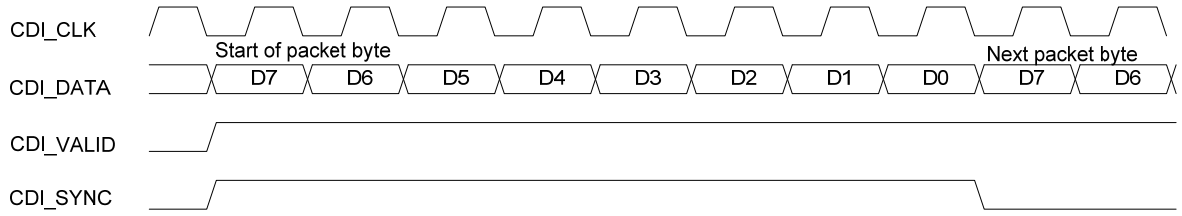


Figure 16. Serial Mode – Continuous Clock

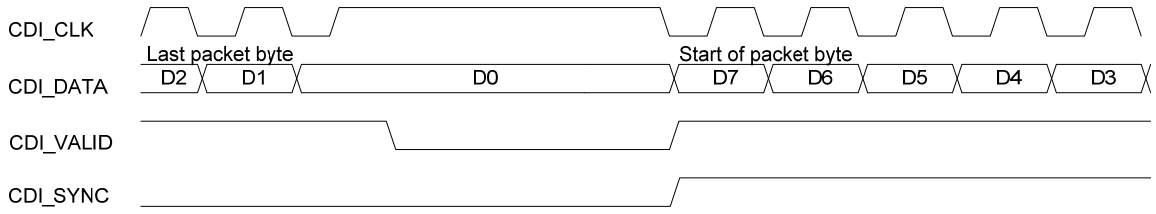


Figure 17. Serial Mode – Gapped Clock

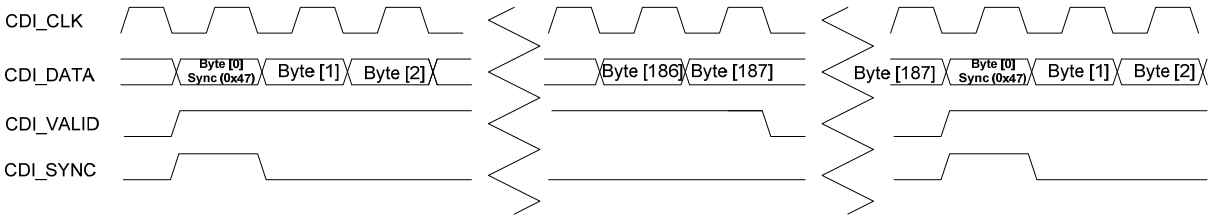


Figure 18. Parallel Mode – Continuous Clock

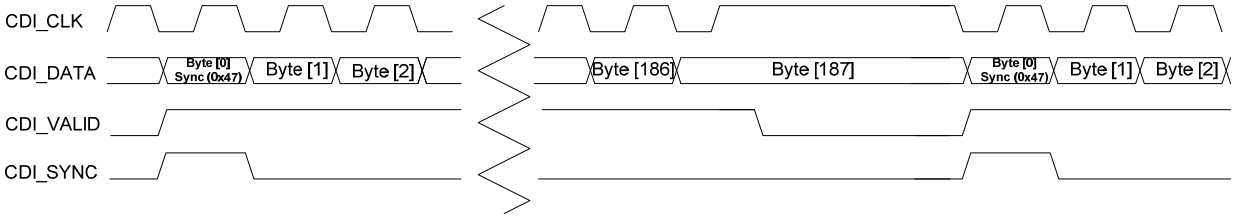


Figure 19. Parallel Mode – Gapped Clock

9 COMPRESSED DATA OUTPUT

The Compressed Data Output (CDO) interface outputs TS data in serial mode to the external decoder.

CDO is designed to interface with a wide variety of devices via a flexible signal configuration. TS data may be partially filtered or processed, for example, SPTS may be output.

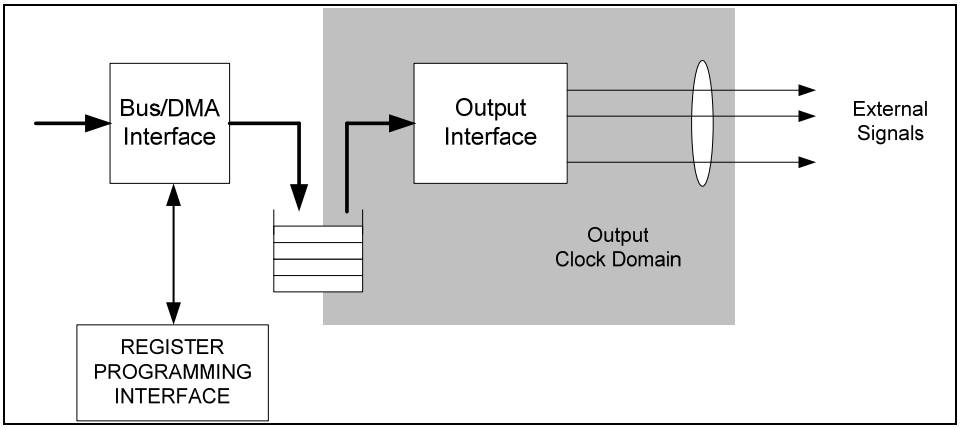


Figure 20. CDO Block Diagram

TS packets are read from DRAM using Data Bus DMA and are stored in CDO FIFO. Data is then serialized and output via serial data interface at the programmable clock rate (up to 100MHz).

CDO signals:

1. TS_CLK (O): (use pin VXI_D23). Clock output.
2. TS_D (O): (use pin VXI_D22). Serial data output.
3. TS_SYNC (O): (use pin VXI_D21). Top/Sync indicator.
4. TS_VALID (O): (use pin VXI_D20). Data Valid.
5. DREQ (I): (use pin VXI_D19). Data Request input signal (optional).
6. TS_ERROR (O): (use pin VXI_D18). Transport Stream Error was detected (optional).

Note: CDO reuses the MSBs of the VXI interface, therefore when CDO is enabled VXI interface becomes 16-bit only.

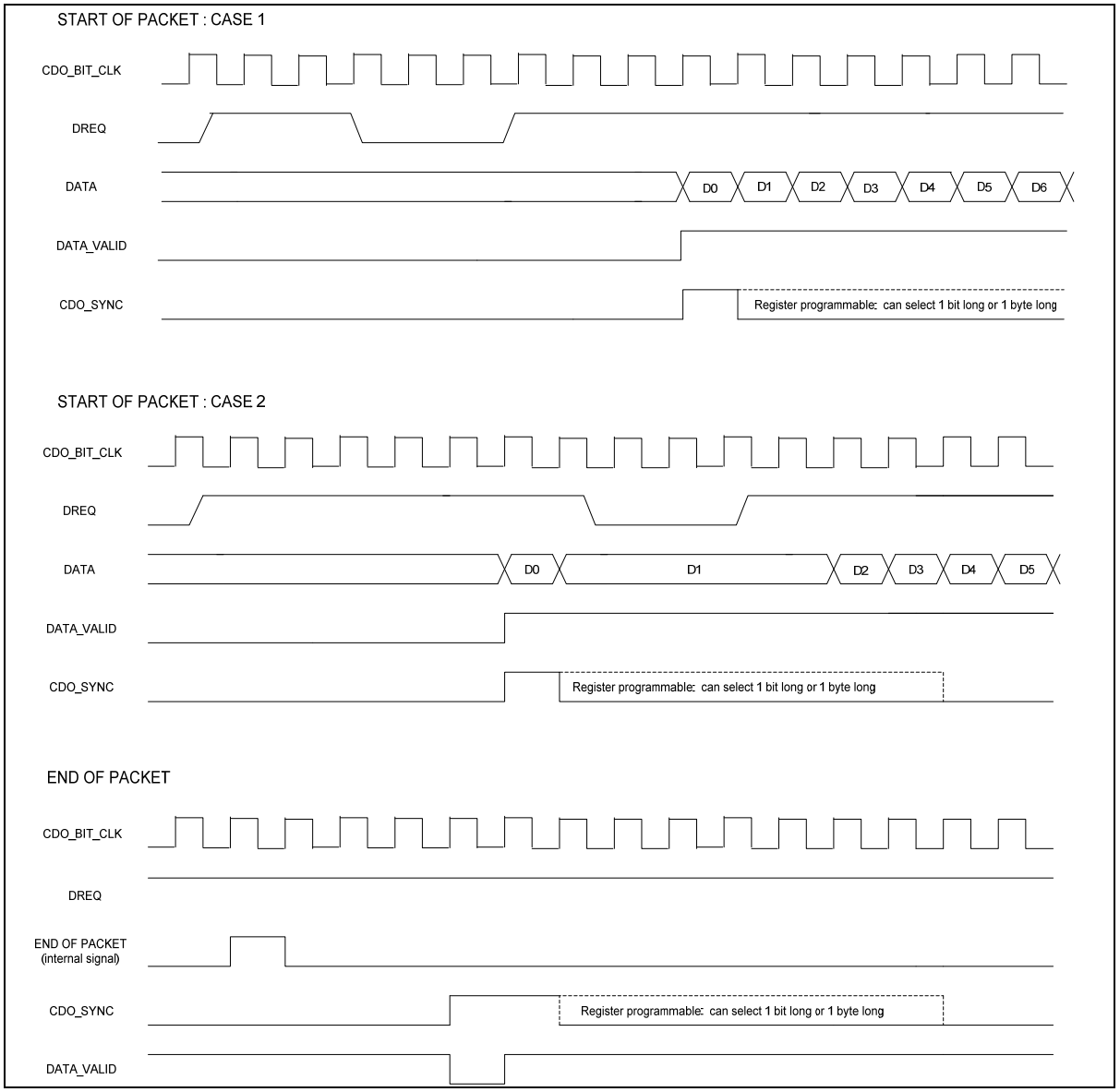


Figure 21. CDO Waveforms

10 MAIN CPU

The MIPS32 4KEc is a 32-bit, five-stage pipeline, Harvard Cache Architecture Processor core complete with 16-bit code compression. The Memory Management Unit (MMU) contains 4-entry instruction and data Translation Lookaside Buffers (ITLB/DTLB) and a 16 dual-entry joint TLB (JTLB) with variable page sizes. The cache in the MIPS core is 4-way set associative and the size of each instruction and data cache is 16 Kbytes.

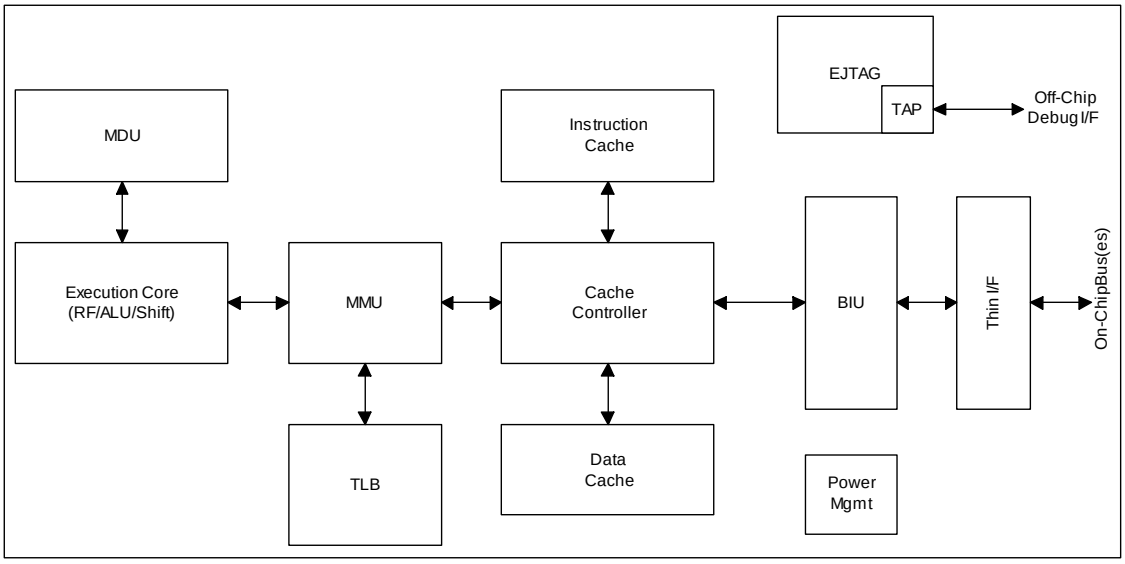


Figure 22. MIPS32 4KEc Block Diagram

Features:

- 5-stage Pipeline
- 32-bit Address and Data Paths
- MIPS32-Compatible Instruction Set
  - Multiply-Accumulate and Multiply-Subtract Instructions (MADD, MADDU, MSUB, MSUBU)
  - Targeted Multiply Instruction (MUL)
  - Zero/One Detect Instructions (CLZ, CLO)
  - Wait Instruction (WAIT)
  - Conditional Move Instructions (MOVZ, MOVN)
  - Prefetch Instruction (PREF)
- MIPS32 Enhanced Architecture (Release 2) Features
  - Support for Vectored interrupts
  - Programmable exception vector base
  - Atomic interrupt enable/disable
  - Bit field manipulation instructions
  - Virtual memory support
- MIPS16e™ Code Compression
- 16-bit encodings of 32-bit instructions to improve code density
  - Special PC-relative instructions for efficient loading of addresses and constants
  - SAVE & RESTORE macro instructions for setting up and tearing down stack frames within subroutines
  - Improved support for handling 8 and 16 bit datatypes

- Cache Sizes
  - I$/D$ size = 4KB x 4-Way Set Associative
  - Loads block only until critical word is available
  - Write-back and write-through support
  - 16-byte cache line size
  - Virtually indexed, physically tagged
  - Cache line locking support
  - Non-blocking prefetches
- MIPS32 Privileged Resource Architecture
  - Count/Compare registers for real-time timer interrupts
  - I and D watch registers for SW breakpoints
- Programmable Memory Management Unit
  - 16 dual-entry JTLB with variable page size
  - 4-entry ITLB
  - 4-entry DTLB
- Multiply/Divide Unit
  - Maximum issue rate of one 32x16 multiply per clock
  - Maximum issue rate of one 32x32 multiply every other clock
  - Early-in iterative divide. Minimum 11 and maximum 34 clock latency (dividend (rs) sign extension-dependent)
- Power Control
  - Minimum frequency: 0 MHz
  - Power-down mode (triggered by WAIT instruction)
  - Support for software-controlled clock divider
  - Support for extensive use of local gated clocks
- EJTAG Debug
  - Support for single stepping
  - Virtual instruction and data address/value breakpoints
  - TAP controller is chainable for multi-CPU debug
  - Cross-CPU breakpoint support

11 COMMUNICATION BUS (COMMBUS)

The Communication Bus allows MPE0, MPE1, MPE2 and MIPS to communicate over a low latency, high-speed bus. This is a 32-bit bus running at the full bus_clock rate. This bus is independent from the DMA D-Bus, and provides an alternative to passing data in memory. This bus is, in many ways, analogous to a simple network.

The Communication Bus is used both as means of inter-processor communication, and as a means of communication with peripherals. The following devices have a Communication Bus interface:

- MPE0-2
- MIPS EBB

Each device attached to the Communication Bus has a transmit buffer, and a receive buffer. Each of these buffers can contain 128 bits of data, along with a Communication Bus address.

11.1 COMMUNICATION BUS IDENTIFICATION CODES

Each MPE is allocated a logical identification code, so that communication is by logical device, rather than physical. Communication Bus identification numbers are allocated 7-bit values as follows:

ID dec	Function
0	MPE0
1	MPE1
2	MPE2
3	MIPS EBB

11.2 DATA TRANSFER PROTOCOL

Before transmitting a packet, users must first ensure that the local transmit data buffer is empty unless it is already empty. Afterwards, the target device address must be written into the Communication Bus control register (if not set up already) and then the transmit data. The act of writing data into the transmit data buffer marks it as full, and initiates the transfer mechanism. The transmit buffer full flag is set until the hardware has transmitted the data, or the transmission fails.

The Communication Bus interface hardware will then request a transfer of data to the selected target by requesting the Communication Bus. The bus is allocated on a round-robin basis between requesting transmitters. When the transmitter is allocated, the bus presents the target ID for the transfer, and its own ID. Two things can then happen. If the target is able to receive data (i.e. its receive buffer is not full or disabled) then the data is transferred over the bus. If the target is unable to receive, the bus transaction terminates. In either case, the transaction is then complete, and the bus is re-arbitrated.

12 HOST INTERFACE

The Host Interface (HIF) provides the boot path for the internal MIPS core. During power up, it interfaces the MIPS to an external Flash device containing boot code. It also implements functionality, together with DMA, for transferring data between external devices interfaced to HIF and the DDR memory.

The host interface supports the following devices:

- 8-bit or 16-bit Boot Flash (NOR)
- NAND Flash
- SPI Flash
- Ethernet controllers
- CableCARD/DVB-CI Conditional Access Modules

HIF provides six chip selects, to enable connection to several devices and supports programmable configuration for each of the chip select signals. Each Chip Select is associated with an address range and is automatically asserted by Host Interface logic for reads/writes in that range. Some of the parameters that can be configured for this interface are:

- Data bus width (8 or 16 bit)
- Polarity of control signals (RD, WR, CS, ACK)
- Protocol (RD/WR asserted first, fixed/variable latency)
- Device Access Timing

All transactions on the host interface follow the big-endian rule. The interface provides one dedicated chip select line for the boot flash, two dedicated chip select lines for interfacing with CableCARD/DVB-CI modules, and the remaining chip selects are used for connecting to other chips like Ethernet controller, 1394 devices, etc.

12.1 HOST INTERFACE COMPONENTS

The host interface consists of the following four major functional blocks (see Figure 23):

- Bus Interface
- MIPS Access Path
- Datamover
- External Interface

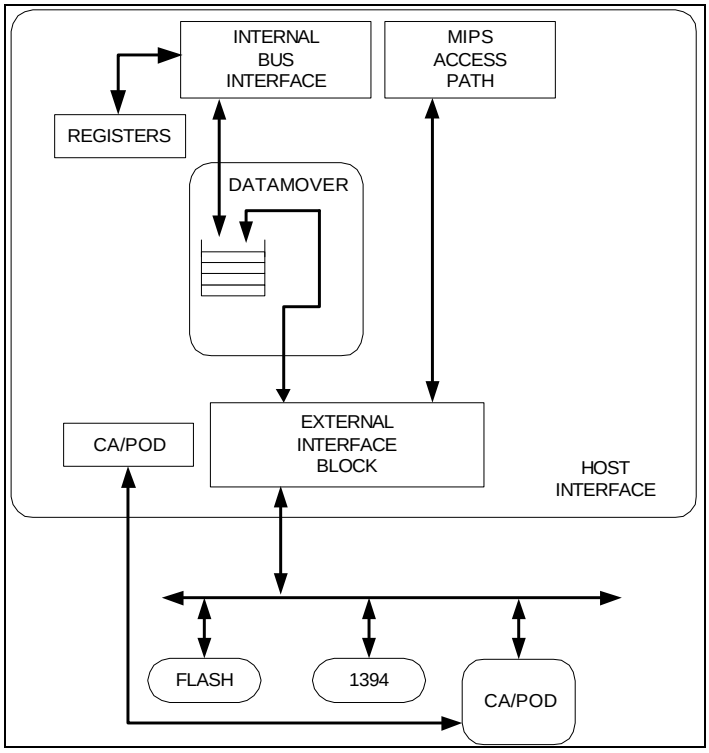


Figure 23. Host Interface Block Diagram

12.1.1 BUS INTERFACE

The bus interface provides access to the internal bus. It also provides a path for the DMA to transfer data from an external device to memory through the Datamover block.

12.1.2 MIPS ACCESS PATH

This interface provides a direct path from MIPS to an external boot flash. A space of 32 MB is reserved in the memory map of the device for a boot FLASH device. MIPS uses this path to read boot code after the de-activation of the reset signal. Access to this interface is enabled only in the FLASH address range. Byte enable is supported for MIPS write access to allow for 8-bit or 16-bit writes.

12.1.3 DATAMOVER

The Datamover block provides a function to transfer a block of data from the external device to memory via the DMA interface.

12.1.4 EXTERNAL INTERFACE

The External Interface block responds to commands from either the bus interface or MIPS and performs either read or write operations. The External Interface block provides the following features:

- Read and write accesses with the selected chipselect asserted
- 8-bit or 16-bit wide access
- Fixed width or extended access via ext_ack input
- Support for programming the polarity of control signals
- Support for programming different access timings

The host interface does not support multiplexed address/data mode.

12.2 HOST INTERFACE TRANSFERS

The host interface supports the following transfer types:

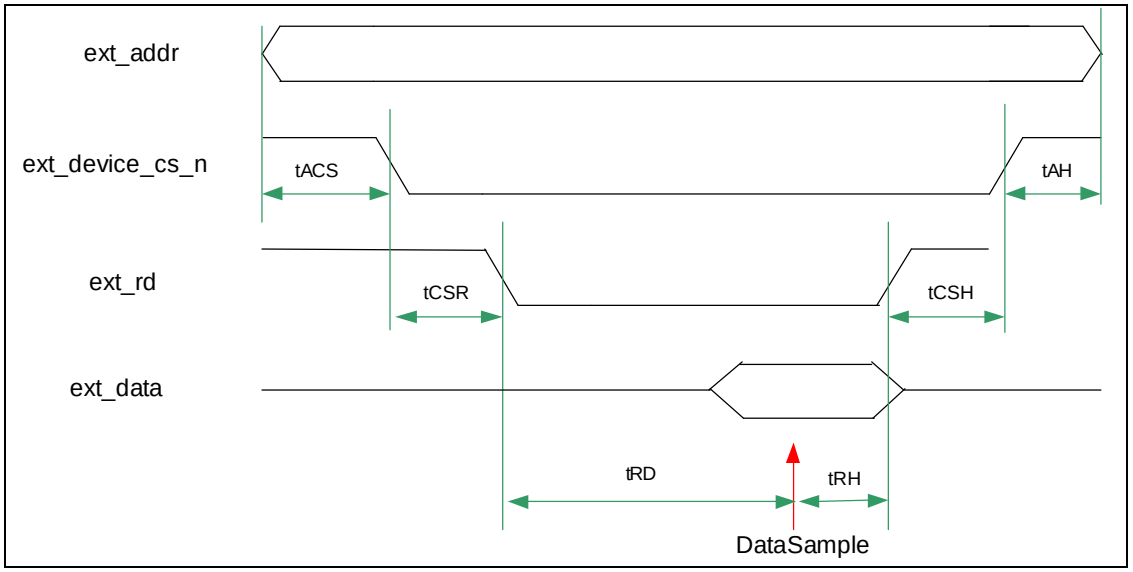


Figure 24. Fixed Latency Read Access – CS Asserted First

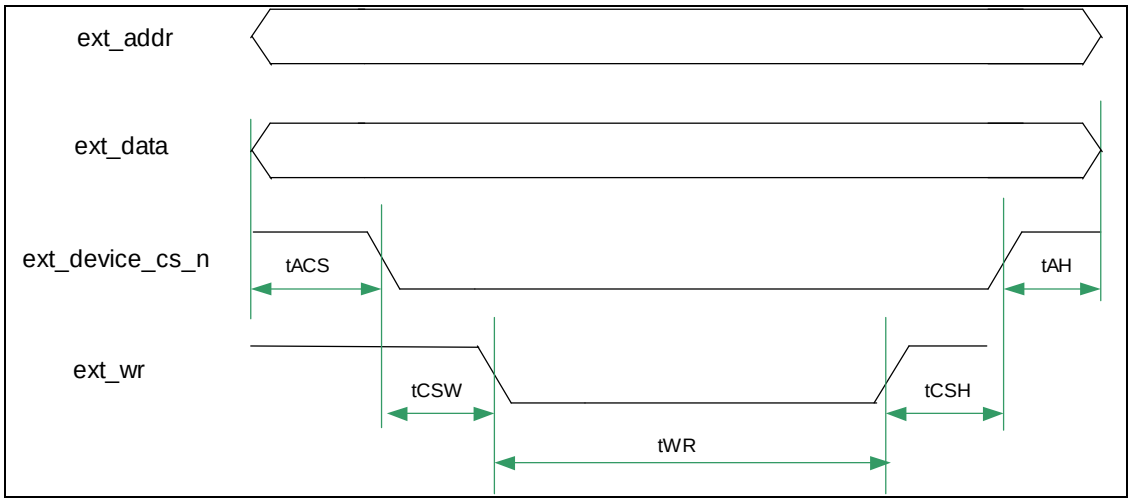


Figure 25. Fixed Latency Write Access – CS Asserted First

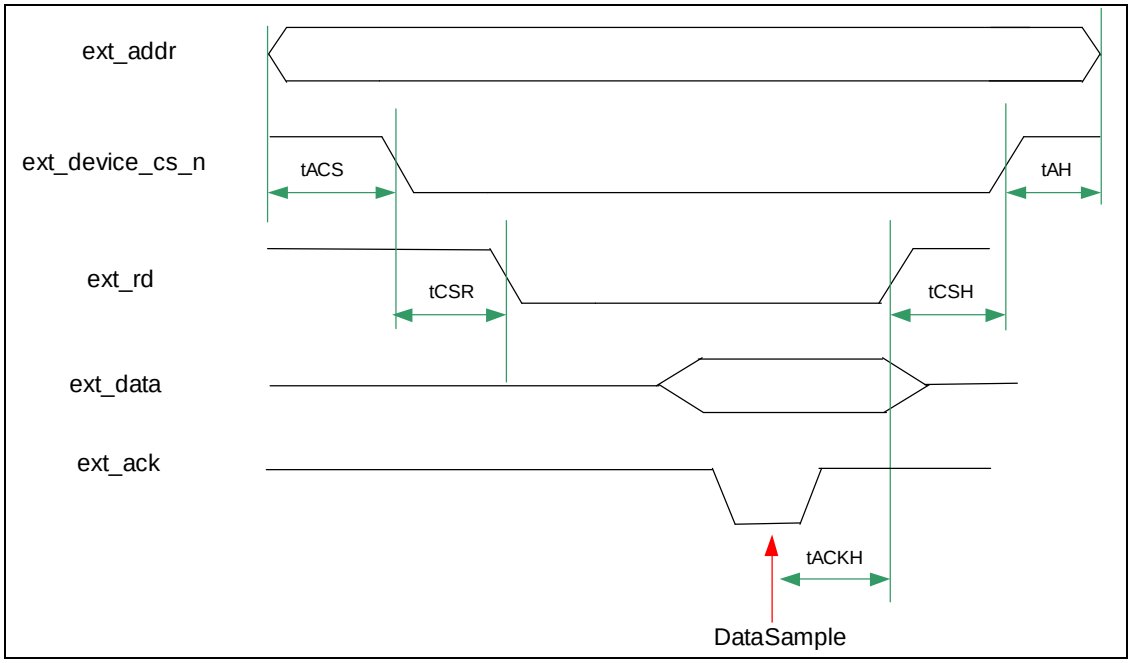


Figure 26. ACK-Driven Read Access – CS Asserted First

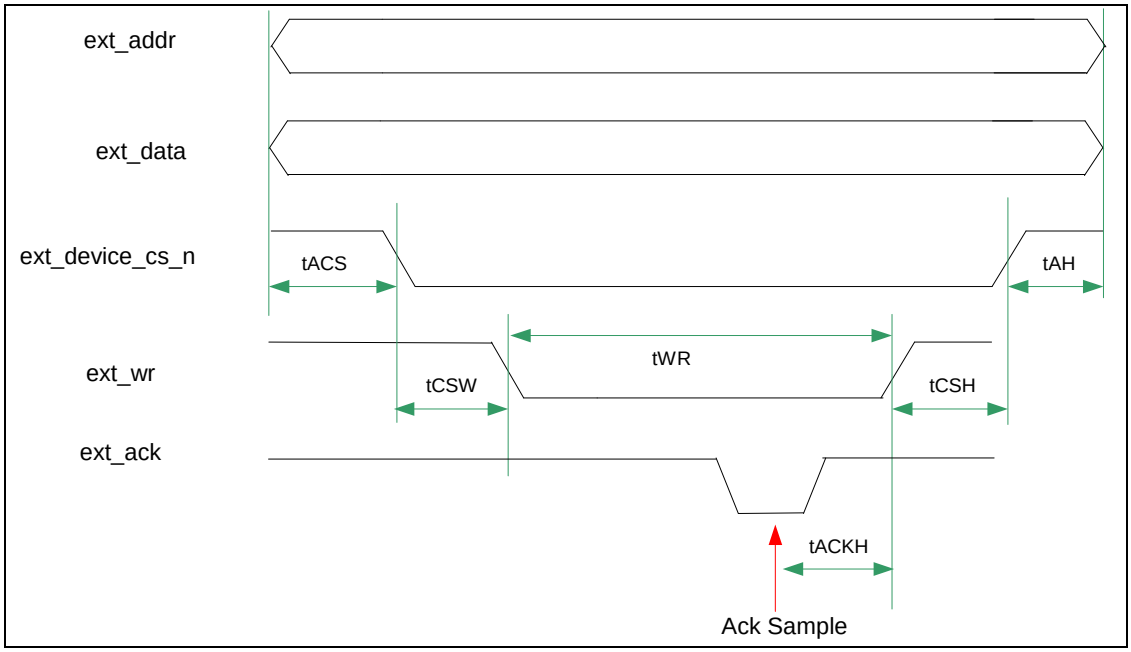


Figure 27. ACK-Driven Write Access – CS Asserted First

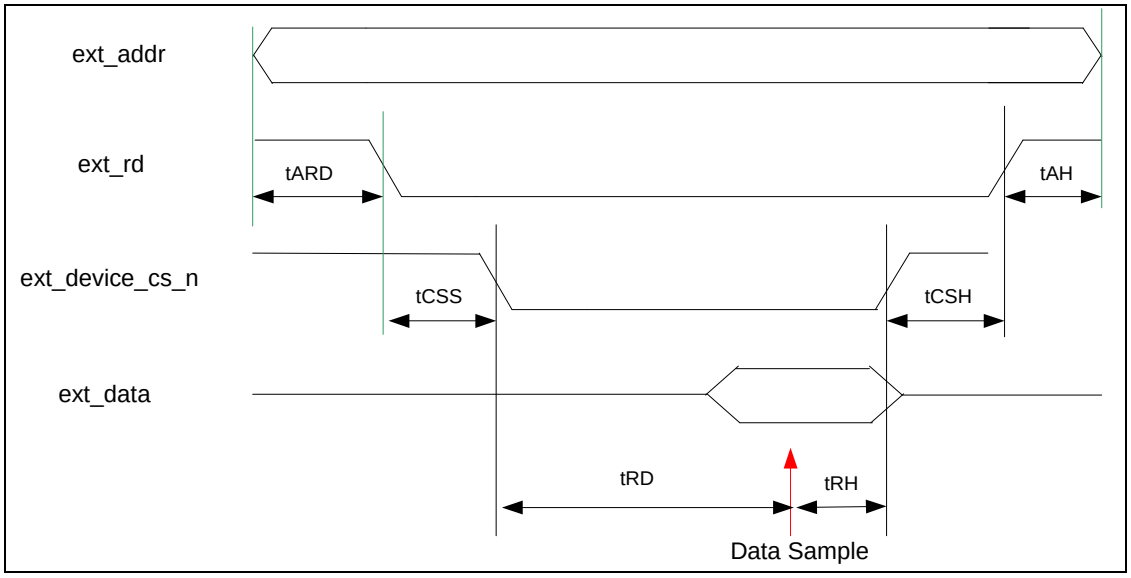


Figure 28. Fixed Latency Read Access – RD Asserted First

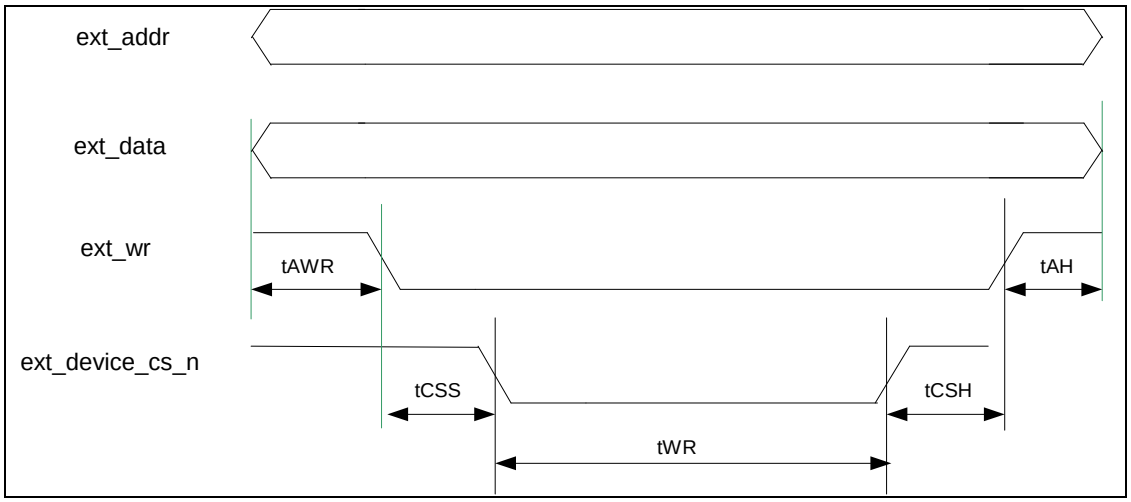


Figure 29. Fixed Latency Write Access – WR Asserted First

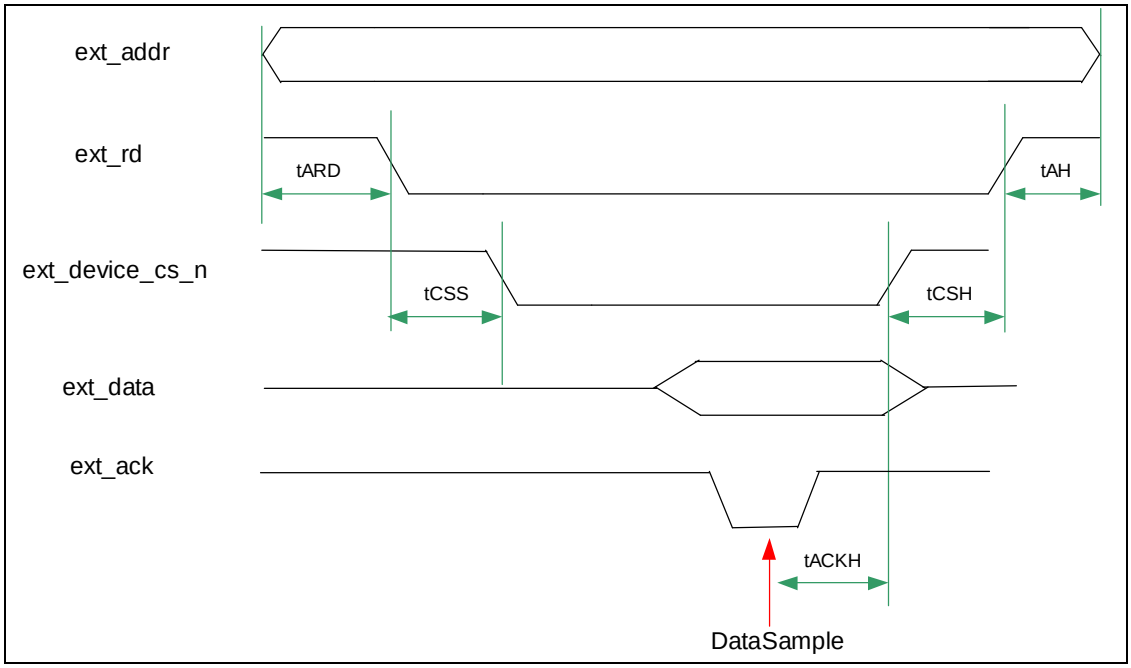


Figure 30. Variable Width Read Access – RD Selected First

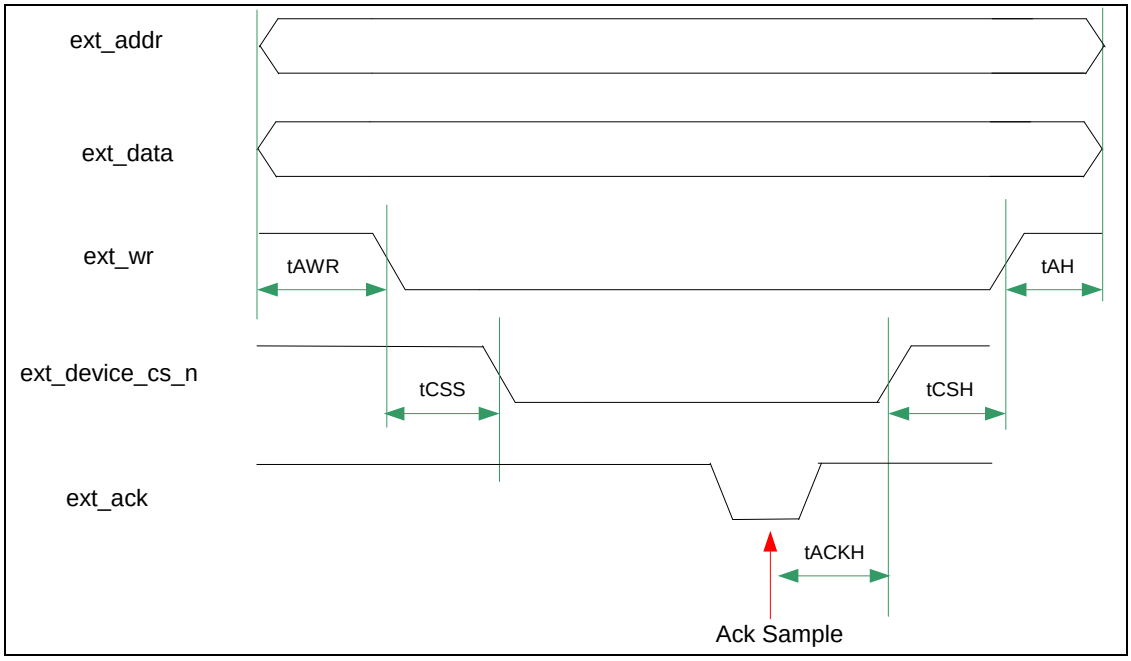


Figure 31. Variable Width Write Access – WR Selected First

Table 29. Definition of Timing Parameters

Parameter	Description
tACS	Address/write data valid to chipselect asserted
tCSR	Chipselect asserted to read enable asserted
tRD	Read enable asserted to data sample
tRH	Data sample to read enable deasserted
tCSH	Read/write enable deassert to chipselect deassert
tAH	Chipselect deassert to Address/write data invalid
tCSW	Chipselect asserted to write enable asserted
tACKH	ACK sampled to read/write deasserted
tARD	Address valid to read enable asserted
tAWR	Address valid to write enable asserted
tCSS	Read/write asserted to chipselect asserted
tCSH	Chipselect deasserted to read/write deasserted

12.3 NAND FLASH SUPPORT

NAND flash can be supported by connecting the NAND flash signals to the appropriate signals in the host bus interface as shown in the table below.

Table 30. NAND Flash and Host Interface Signal Mapping

NAND Flash Signal	Host Interface Signal
IO[0..7]	POD_HOST_D[0..7]
RE	POD_OE_HOST_RD
WE	POD_WE_HOST_WR
ALE	POD_HOST_A[1]
CLE	POD_HOST_A[2]
R/B_n	Any GPIO pin
CS	HOST_CS_0_N

Both Small Block and Large Block NAND Flash are supported.

HIF implements ECC error detection for Small Block NAND Flash. To use ECC calculation, you must read all 528 Bytes (512 data + 16 extra) of the NAND page, using one DMA command; then 1-bit and 2-bit errors can be detected. 1-Bit error is reported in the ECC_ERROR_ADDR register.

12.4 CA/POD

The FLI106xx chip has an integrated CableCARD/DVB-Common Interface controller that provides an almost glueless interface with a CableCARD/DVB-CI connector (similar to a PCMCIA connector). The CableCARD interface conforms to the OC-SP-CCIF2.0-I10-070323 standard and supports S-Mode cards only. The Common Interface is compliant with the EIA-679-Part B standard. To minimize the pins required to support the wide PCMCIA connector, some of the pins of this interface are shared with the host bus signals.

The built-in CableCARD/CI controller with the associated glue-logic is capable of the following functions:

- Automatically detecting the insertion/removal of the card (CableCARD/CI) with debounce logic. The card detect logic can be triggered by software as well.
- 3.3V Hot-insertion using external buffers.
- Automatically re-configuring the pins of the interface to memory or CableCARD/CI mode (after the personality change) after a hard or soft host reset.

Table 31 below shows the state of the FLI106xx CableCARD/CI signals before and after a personality change in CableCARD and CI modes.

Table 31. FLI106xx CableCARD/CI Signals

PCMCIA Connector Pin #	PCMCIA Signal Name	FLI106xx Signal	Before Personality Change	After Personality Change	
				CableCARD	Common Interface
1	Ground				
2	D3	POD_HOST_D3*	D3	D3	D3
3	D4	POD_HOST_D4*	D4	D4	D4
4	D5	POD_HOST_D5*	D5	D5	D5
5	D6	POD_HOST_D6*	D6	D6	D6
6	D7	POD_HOST_D7*	D7	D7	D7
7	CE1#	POD_CE_1	CE1#	CE1#	CE1#
8	A10	POD_HOST_A10*	A10		A10
9	OE#	POD_OE_HOST_RD	OE#	OE#	OE#
10	A11	POD_HOST_A11*	A11		A11
11	A9	POD_A9_DRX	A9	DRX	A9
12	A8	POD_A8_CRX	A8	CRX	A8
13	A13	POD_HOST_A13*	A13		A13
14	A14	POD_A14_MCLKO	A14	MCLKO	A14
15	WE#	POD_WE_HOST_WR*	WE#	WE#	WE#
16	READY	POD_READY_IRQ_N	READY	IRQ#	IRQ#
17	3.3V				
18	5V/3.3V			5V/3.3V(1)	VPP1
19	A16		Not Used	MIVAL	MIVAL
20	A15		Not Used	MCLKI	MCLKI
21	A12	POD_HOST_A12*	A12		A12
22	A7	POD_A7_QTX	A7	QTX	A7
23	A6	POD_A6_ETX	A6	ETX	A6
24	A5	POD_A5_ITX	A5	ITX	A5
25	A4	POD_A4_CTX	A4	CTX	A4
26	A3	POD_HOST_A3*	A3	A3	A3
27	A2	POD_HOST_A2*	A2	A2	A2
28	A1	POD_HOST_A1*	A1	A1	A1
29	A0	POD_HOST_A0*	A0	A0	A0
30	D0	POD_HOST_D0*	D0	D0	D0
31	D1	POD_HOST_D1*	D1	D1	D1
32	D2	POD_HOST_D2*	D2	D2	D2
33	WP				HIGH
34	Ground				

PCMCIA Connector Pin #	PCMCIA Signal Name	FLI106xx Signal	Before Personality Change	After Personality Change	
				CableCARD	Common Interface
35	Ground				
36	CD1#	POD_CD_1	CD1#	CD1#	CD1#
37	D11	POD_D11_MDO3	Not Used	MDO3	MDO3
38	D12	POD_D12_MDO4	Not Used	MDO4	MDO4
39	D13	POD_D13_MDO5	Not Used	MDO5	MDO5
40	D14	POD_D14_MDO6	Not Used	MDO6	MDO6
41	D15	POD_D15_MDO7	Not Used	MDO7	MDO7
42	Extended Channel	POD_CE_2	CE2#	CE2#	CE2#
43	VS1#		Hi-Z		VS1#
44	RFU	POD_IORD_HOST_A6*		IORD#	IORD#
45	RFU	POD_IOWR_HOST_A5*		IOWR#	IOWR#
46	A17		Not Used	MISTRT	MISTRT
47	A18		Not Used	MDI0	MDI0
48	A19		Not Used	MDI1	MDI1
49	A20		Not Used	MDI2	MDI2
50	A21		Not Used	MDI3	MDI3
51	3.3V				
52	5V/3.3V			5V/3.3V(1)	VPP2
53	A22		Not Used	MDI4	MDI4
54	A23		Not Used	MDI5	MDI5
55	A24		Not Used	MDI6	MDI6
56	A25		Not Used	MDI7	MDI7
57	VS2#	POD_VS2_MCLKO	Hi-Z		MCLKO
58	RESET	POD_RESET	POD_RESET	POD_RESET	POD_RESET
59	WAIT#	POD_WAIT_N	WAIT#	WAIT#	WAIT#
60	INPACK#				INPACK#
61	REG#	PODREG_HOST_A4*	REG	REG	REG
62	BVD2#	POD_BVD2_MOVAL	BVD2	MOVAL	MOVAL
63	BVD1#	POD_BVD1_MOSTRT	BVD1	MOSTRT	MOSTRT
64	D8	POD_D8_MDO0	Not Used	MDO0	MDO0
65	D9	POD_D9_MDO1	Not Used	MDO1	MDO1
66	D10	POD_D10_MDO2	Not Used	MDO2	MDO2
67	CD2#	POD_CD_2	CD2#	CD2#	CD2#
68	Ground				
		POD_DIR_N			
		POD_DETECT_N			
		OOB_CTX			
		OOB_CRX			
		OOB_DRX			

* Signals shared with the host interface.

(1) The host can dynamically switch the supply for this pin between 3.3V and 5V, based on the CIS information of the CableCARD. By default without CableCARD plugging in, the voltage is 3.3V.

12.5 SPI FLASH

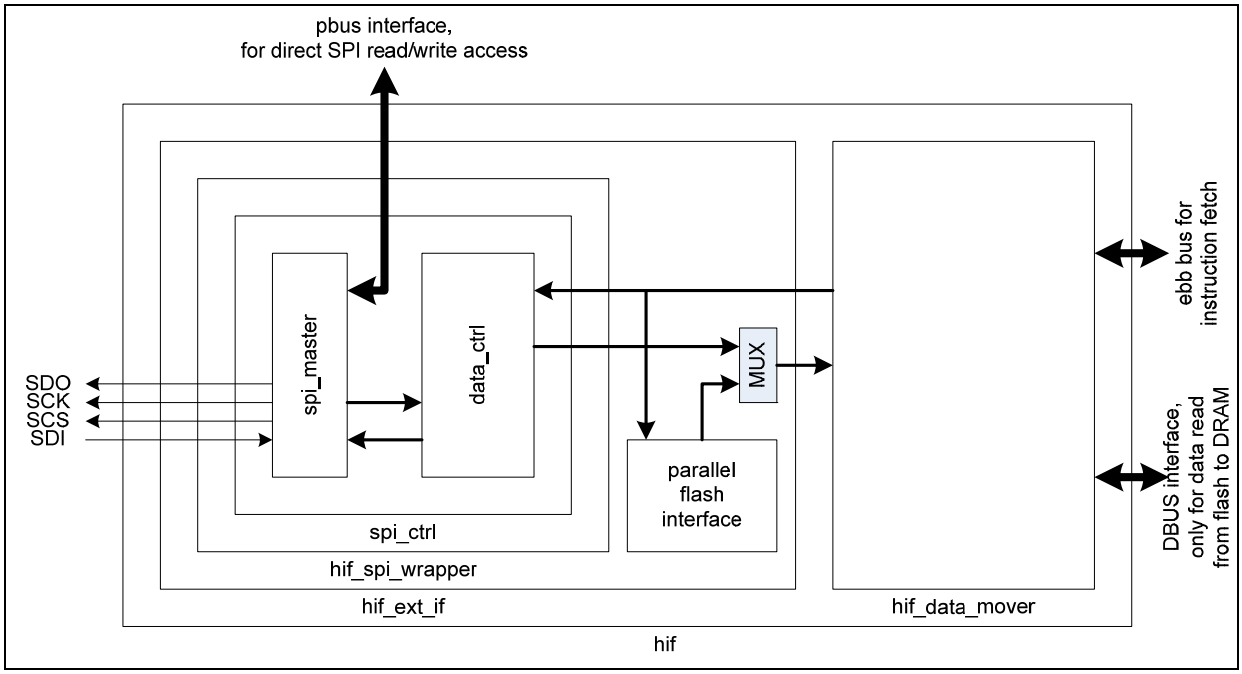


Figure 32. SPI Flash

Hardware support is provided for SPI Serial FLASH ROM up to 4 M at a minimum speed of 33, 40, 50, 67, and 100 MHz. The SPI interface is configured as follows:

1. SPI_CS may be connected to any one of: HOST_BOOT_CS_N, HOST_DEV_CS0_N, HOST_DEV_CS1_N,, HOST_DEV_CS2_N, POD_CE_1, POD_CE_2. Controlled by bit HIF_SPI_CONTROL.SPI_CS_SEL.
2. SPI_CLK (output) – POD_HOST_A3
3. SPI_SDO (output) – POD_HOST_A2
4. SPI_SDI (input) – POD_HOST_D0

The SPI Flash controller supports direct MIPS CPU boot from SPI Flash. For boot from SPI bit BOOTSTRAPS.BSTRAP_BOOT_MODE must be set to '10' and BOOTSTRAPS.BSTRAP_NOR_FLASH_SEL must be set to '1'.

The SPI Flash controller also supports fast read mode using DMA. MIPS CPU can also directly read or write single byte/16-bit half-word data.

13 MEDIA PROCESSING ELEMENTS (MPES)

The FLI106xx has three Media Processing Elements (MPEs). Each MPE is a fully independent, variable-length Very-Long-Instruction-Word (VLIW) processor and has five distinct functional units:

- **Arithmetic Logic Unit (ALU):** The ALU essentially consists of a 32-bit Arithmetic Operation Unit (AOU), with a variety of pre-processing options on the source data.
- **Multiply Unit (MUL):** The MUL can perform two fundamental operations:
  - 32x32 signed multiply, with a sign extended 32-bit result extracted by an appropriate arithmetic shift
  - Four independent 16x16 signed multiplies with four 32-bit results with a range of shift options
- **Execution Control Unit (ECU):** The ECU is responsible for controlling the program counter and execution pipeline.
- **Register Unit (RCU):** The RCU is responsible for the control of special purpose registers within the MPE.
- **Memory Unit (MEM):** The MEM is responsible for data transfers between internal MPE registers and data memory.

In a High-Definition decode application, each MPE is configured for a particular task (i.e. MPE0 is configured for audio decode, MPE1 for transport stream de-multiplexing, and MPE2 for MPEG video decoding).

14 TRANSPORT DE-MULTIPLEXER

The FLI106xx chip supports de-multiplexing of one High Definition MPEG stream or up to two Standard Definition MPEG streams (as specified in the ATSC A/53C compression format table). The de-multiplexer engine parses the incoming compressed transport stream and routes the extracted video, audio, and associated data packets into memory.

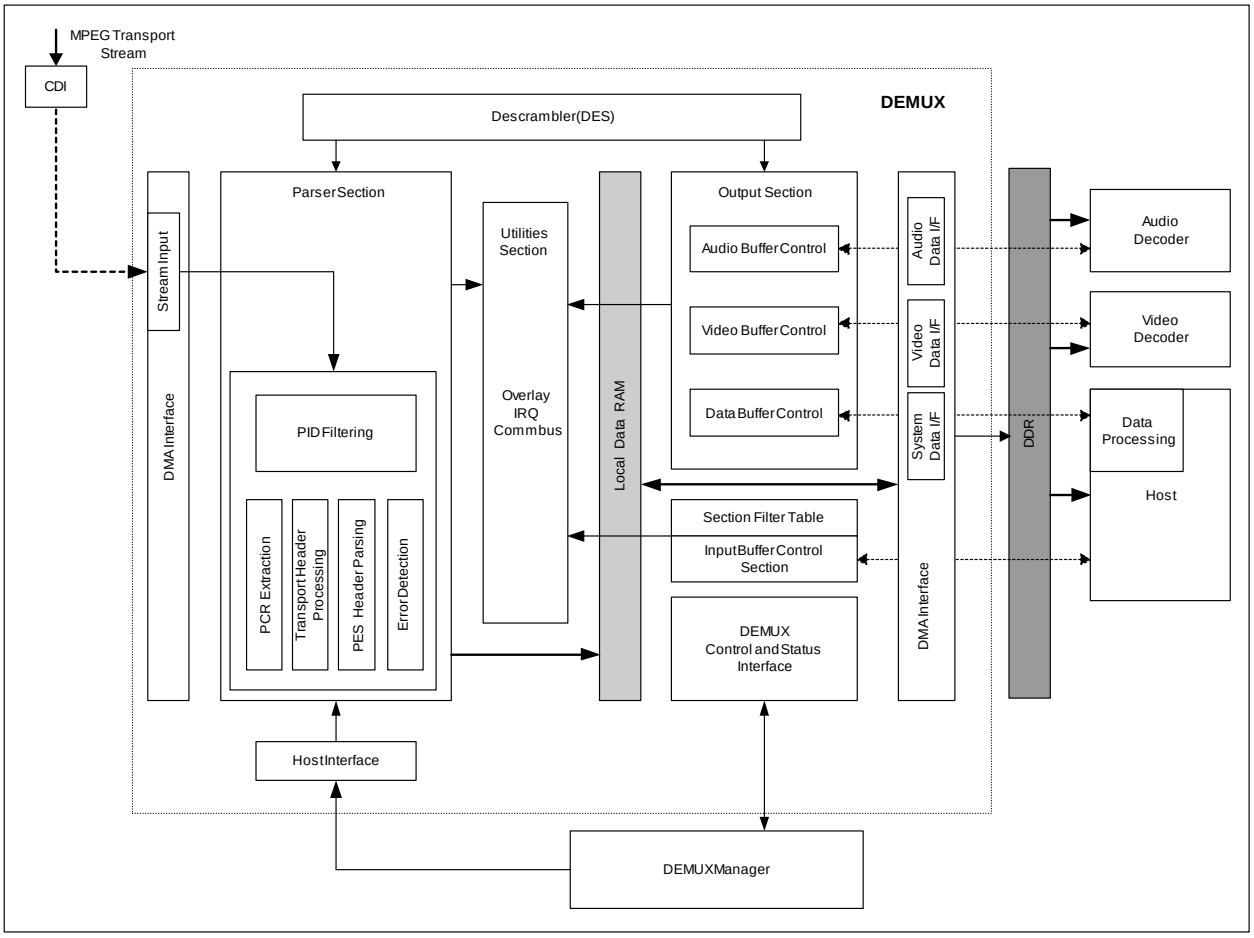


Figure 33. De-Multiplexer Architecture

The de-multiplexer supports configuring and maintaining up to 64 Program Identifications (PIDs) and 128 section filters. It supports a maximum aggregate data rate of 80Mbps. Additionally, the de-multiplexer can extract PCR information from adaptation field, monitor continuity counter, monitor scramble bits, and do packet substitution via Display Index Table (DIT) and Service Information Table (SIT).

14.1 FUNCTIONAL DESCRIPTION

Demux is primarily implemented as a software component; the only hardware component in demux is the DES processor. The demux block runs on MPE-1 and receives input from the CDI block. CDI syncs up the transport packets and then internally buffers the transport stream data in its FIFO. A ping-pong DMA buffer is set up for moving the transport stream data from CDI to the MPE. Each transfer consists of one Transport packet length (i.e. 188 bytes).

The demux software architecture consists of two major sections:

- Parser
- Output/Routing

The parser section is responsible for PID processing and header parsing. The demux manager on the host processor (MIPS) configures the filters in demux. Depending on the filter configuration, demux parses the transport stream packet and extracts data. The extracted data can be audio/video, elementary streams, or system data, depending on the filter PID set by the demux manager. The parser section outputs all of the extracted data to an internal memory. This memory interfaces with the audio/video decoders and the host, via the DMA interface. Based on the kind of the selected parser, the utilities section manages the code overlay. The utilities section consists of IRQ routines and CommBus utilities.

The output/routing section is responsible for managing buffers. The output section includes a CommBus interface that manages the control/status flow between demux MPE, MIPS, and audio/video MPE. It sends audio, video, and system data to the respective track buffers (located in external DDR). Demux and the video/audio decoder have a buffer management scheme with a write and read pointer associated with the buffers which sits in the DDR. Demux manages the write pointer and the read pointer is managed by the corresponding decoder. Based on the data received from the navigation routine, MIPS tells Demux which section filter table needs to be captured before assembling transport stream data for de-multiplexing.

The DES processor in demux is responsible for decrypting encrypted transport streams before sending the data for decode. The DES module is implemented as a co-processor to MPE-1. It implements the DES Electronic Code Book (ECB) mode of the Data Encryption Standard (DES) algorithm for protecting broadcast information. When demux receives encrypted data, MIPS provides the 56-bit key to unlock the data packet. The DES processor uses this 56-bit key to decrypt transport streams.

15 MPEG VIDEO DECODING

The FLI106xx MPEG Video Decoder can decode one MP@HL MPEG-2 stream or up to two simultaneous MP@ML MPEG-2 streams. The decoder is capable of decoding all DVB, ATSC™ and OpenCable™ video formats.

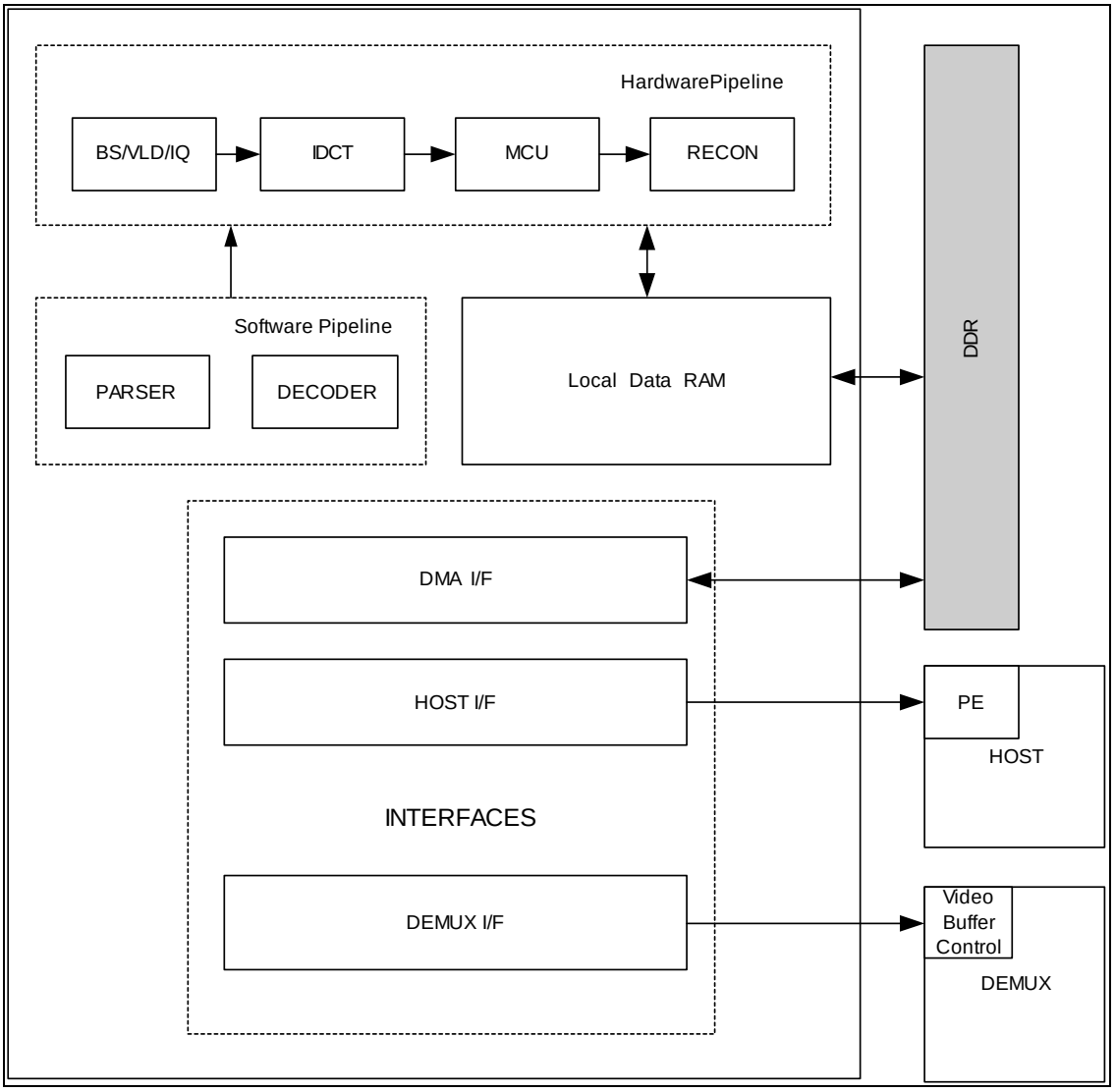


Figure 34. MPEG Video Decoder Block Diagram

15.1 FUNCTIONAL DESCRIPTION

The MPEG decoder firmware is designed to make optimal use of the double buffer feature of the hardware. The firmware supports MPEG processing by implementing the following functionality:

- **Initialization:**
  - Initializing the software and the hardware during cold start and whenever there is a change of MPEG decoder.
- **DMA Management:**
  - Creating and issuing DMA requests to drive the supply of bits from the video bit stream into the MPEG coprocessor.
  - Generating and issuing DMA requests to write each reconstructed macroblock out of the MPEG coprocessor into the output frame buffer.
  - Responding to DMA interrupts, stage-done interrupts, and error conditions.
- **Parsing:**
  - All high-level headers (picture level and above) from the bit stream.
  - Parsing slice and macroblock headers.
- **Co-Processor Management:**
  - Constructing and programming the registers that control the different stages of the MPEG coprocessor.
  - Controlling stage transitions in the MPEG coprocessor pipeline and monitoring the pipeline for error conditions.
- **Error Handling:**
  - Responding to hardware-detected errors, detecting other errors in software, recovering from all errors, and attempting error concealment where feasible.
- **Buffer Management:**
  - Coordinating with the demux, which is supplying bits to the video elementary bit stream buffer.
- **Status/Control Information Exchange:**
  - Exchanging operational and status information with the Presentation Engine on MIPS.

15.2 MPEG PROCESSING

The MPEG decoder processes a single MPEG macroblock (MB) in four pipe stages:

1. Bit Shifter/Variable Length Decode/Inverse Quantization (BS/VLD/IQ)
2. Inverse Discrete Cosine Transform (IDCT)
3. Motion Compensation Unit (MCU)
4. Write Back (WB)

Each MB processing pipeline gets its data from the memory and puts its results back to the memory. The double buffered memory architecture allows the data producer to write data to the next pipe stage while the data consumer in the next pipe stage reads from the other data buffer.

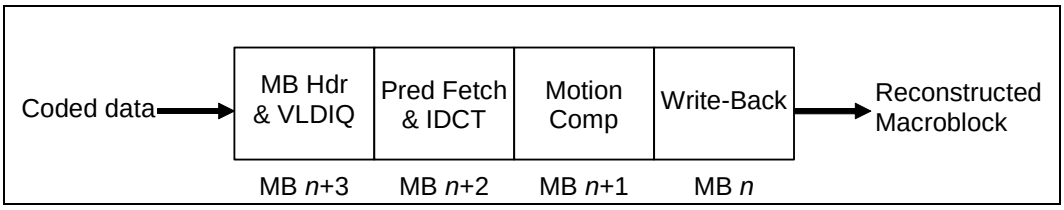


Figure 35. MPEG Decode Sequence

15.2.1 BS/VLD/IQ

The BS/VLD/IQ block assists in faster decoding of MPEG streams. The block provides inverse quantized output by:

1. Providing the next n bits of the coded input stream for n <= 28, advancing the stream by n bits when they are consumed by either the software or the MPEG hardware
2. Rapidly finding the next MPEG start-code
3. Rapidly converting variable-length codes into their decoded values using Table Lookup
4. Decoding 8x8 blocks of variable-length, entropy-encoded, linearized, quantized coefficients of the DCT of the residual signal

15.2.2 IDCT

The Inverse Discrete Cosine Transform (IDCT) is performed on the Inverse Quantized residual coefficients obtained from the previous block of the co-processor. The inverse DCT operation is performed on each of the six blocks of a macroblock.

15.2.3 MCU

Blocks of reference macroblocks are transferred to the predictor buffers using DMA. Motion compensation is performed on the current macroblock. The co-processor does the interpolating and averaging of motion predictors and combining them with the residual signal. The output of the motion compensation unit is stored in recon_mem buffer.

15.2.4 WB

In the WB stage, the reconstructed macroblock from recon_mem is stored to memory in the proper location of the picture being decoded. The DMA Controller performs this write-back to the off-chip memory.

15.2.5 MPEG CLOCK RECOVERY

The MPEG Systems Layer provides a mechanism for the device to remain in step with the studio encoder when both are working in real time. When the encoder is processing data for transmission, a video-locked master clock drives a counter whose value is periodically multiplexed into the transport stream as the Program Clock Reference (PCR). The PCR value is transmitted without passing through any of the large elementary stream encoder buffers.

MPEG clock recovery is handled by the following components:

- PCR extract Logic in CDI (extracts PCR from the incoming TS packets)
- Timers running on the recovered 27 Mhz Clock
- Extended Direct Digital Synthesis (EDDS) modules
- External VCXO (software programmable via the PWM interface)

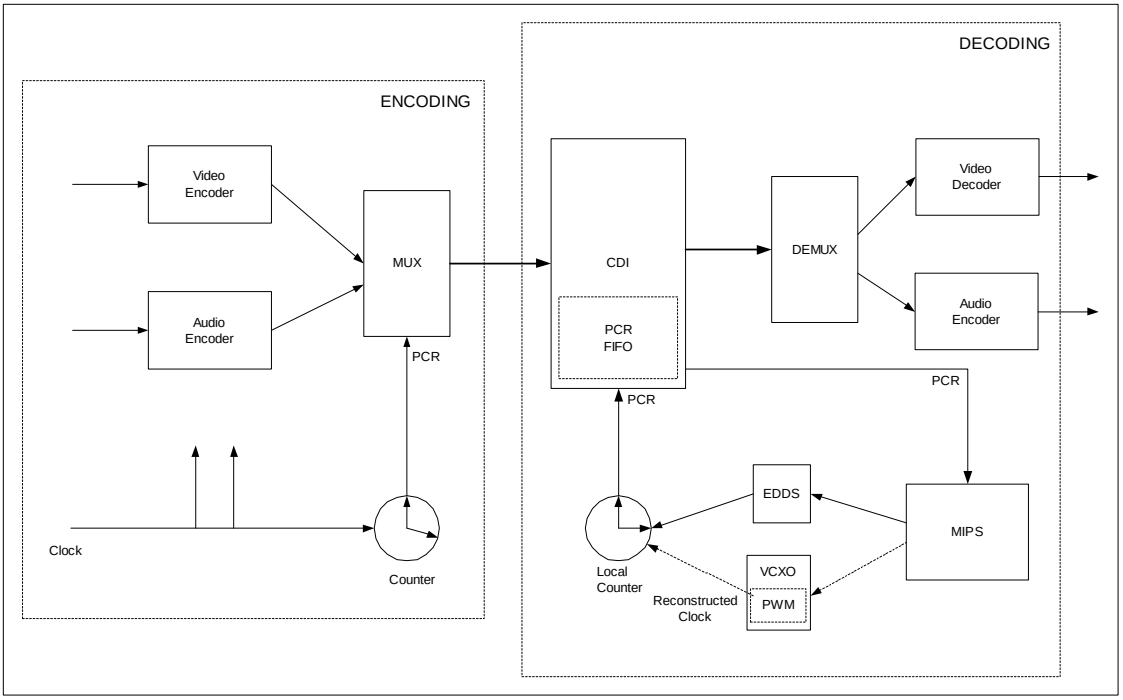


Figure 36. MPEG Clock Data Recovery Process

The CDI block extracts PCR values from the incoming transport stream packets and puts these values into a FIFO. The software then reads the extracted PCR values and determines whether the local 27 MHz clock needs to be sped up or slowed down in order to match it with the transmitter clock. The MIPS applies the correction by programming the EDDS output frequency through register programming.

15.3 MPEG DISPLAY TIMING PROCESSOR

MPEG Display Timing Processor (MDTP) receives MPEG decoded field/frame pixels from the 64-bit DMA data bus (Dbus) and transfers the data to the video backend in programmable raster format. The MDTP data path is divided into luma and chroma data paths. Luma channel simply transfers luma data from Dbus to the backend. Chroma channel has a vertical scaler to perform 4:2:0 to 4:2:2 conversion before sending the data to the backend. MDTP receives input data by generating requests to the DMA controller. It supports 4:2:0, 4:2:2 16-bit video input, and 4:2:0 to 4:2:2 scaling in the chroma data path.

16 DIGITAL AUDIO DECODE

The Multi Audio Decoder (MAD) is a framework that supports the selection of the audio decoder (e.g. Dolby Digital AC3, MPEG, MP3, MP3Pro, and WMA) and the post processing algorithm supported by FLI106xx. Each individual decoder is a piece of low level software that converts compressed audio data into multi-channel PCM samples.

The Dolby Digital AC3 decoder is capable of decoding five full bandwidth channels and one low frequency channel encoded at a sampling frequency of 48 kHz. FLI106xx also supports decoding of emergency broadcast AC3 audio channel.

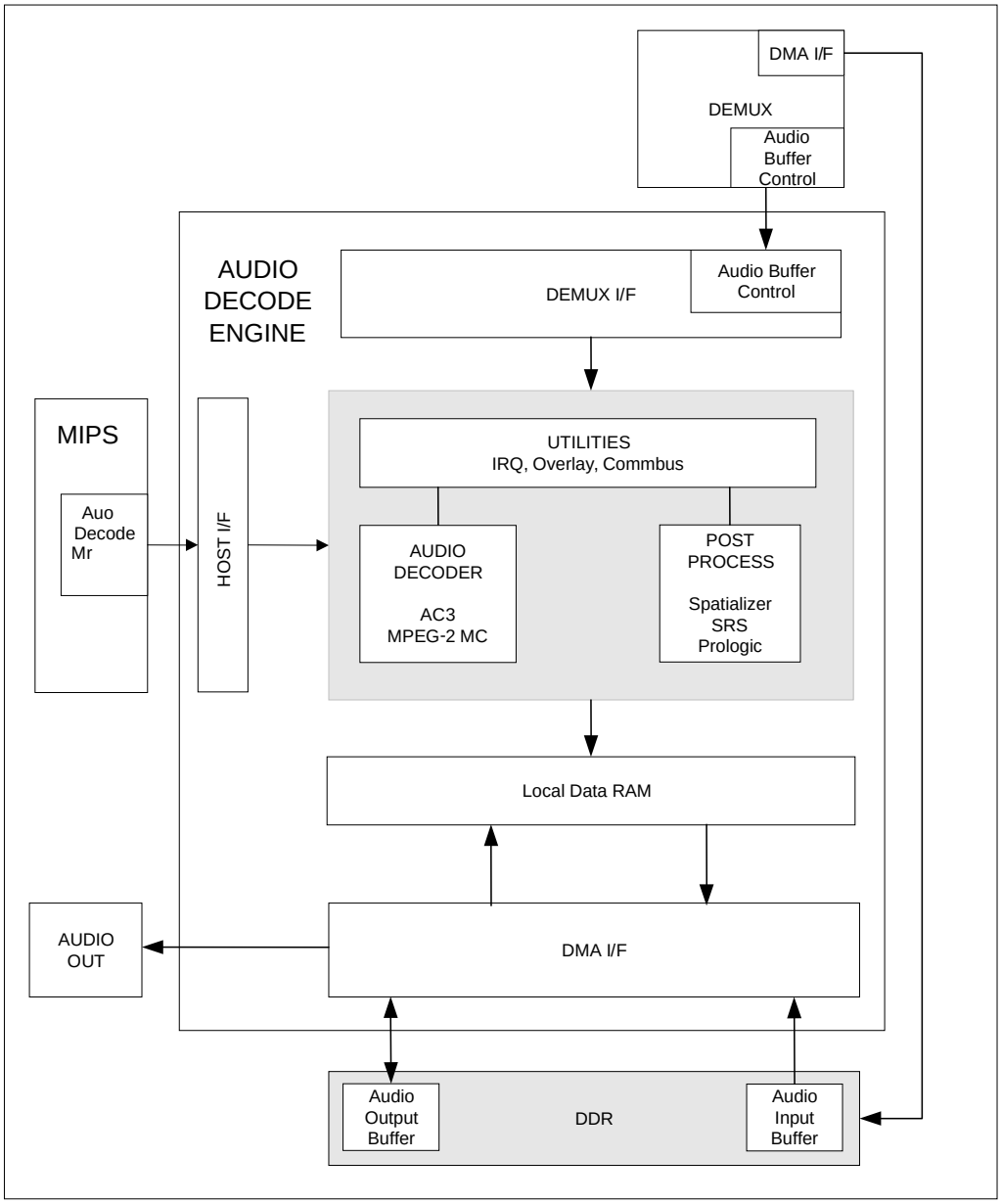


Figure 37. Audio Decoder Block Diagram

The MAD module is a generic design, capable of handling future requests for additional decoding modules or post process modules. MAD consists of the following blocks:

- A decoding engine – can be an AC3 decoder, an MPEG-MC decoder, or any other audio decoder.
- A utility module with basic IRQ routines for user inputs from MIPS (audio stop, start, pause etc.), CommBus routines to communicate with Demux and PE (MIPS), and data/code overlay routines for overlay mechanism for decoder overlay post process overlay.
- A post-processing module for additional post-processing on the decoded audio to improve the quality of sound. Modules like SRS True Surround, spatializer N-2-2, Dolby Prologic, Karaoke, and PCM overlays are some of the post processes.

In addition to these blocks, the decoder includes the following main interface modules which connect to external peripherals and processors for data/control exchanges:

- **Demux Interface:** The demux interface uses the CommBus utilities to manage the set of audio encoded data buffers received by the CDI interface of demux.
- **Host Interface:** The primary controller in the DTV system is MIPS. The host interface provides information to the audio decoder about the appropriate firmware and post-process routine that needs to be loaded for specific channels.
- **DMA Interface:** Encoded audio buffers and decoded audio buffers move from the local scratchpad memory to DRAM and vice-versa. Movement of this data happens over the DMA interface. In addition, the DRAM is used for loading the code overlay to the MPE for specific post-processing algorithms.

16.1 FUNCTIONAL DESCRIPTION

The audio decode block interacts with MIPS via the host interface. The transport stream data received by the CDI provides information on the audio format. MIPS selects the audio data format by reading the transport stream header and then communicates this information to the audio decode block via the CommBus interface.

The decoding framework provides the user the option of configuring various post-processing that can be done on the decoded audio data. The Utilities module of the audio decoder activates the specific firmware after each 256 samples of audio data has been completely decoded. The decoded data is then stored in the DDR in the standard PCM format.

The AC3 decoder supports dynamic range, downmixing, bass redirection, and Karaoke features. These are known as Optional Post Processes.

The speaker and volume settings can be configured by MIPS (user interface) or set to default values. These set of post processes are called Output Post Processes. Output Post Processes comprise the final level of post-processing that is applied on decoded data before channeling the data to an Audio out interface.

The final decoded data after the post process is stored in the DDR and DMAs are setup for the Audio out hardware.

Genesis will support the Audio Description feature under DVBT (for dual audio decode).

17 AUDIO INPUT

FLI106xx has a multi-standard TV Sound IF (SIF) demodulator and audio decoder based on the MPE DSP engine that is designed to be a highly integrated solution for flat panel/pixilated TVs. This audio processing block is a multi-standard audio processor that covers the audio processing of all analog TV standards worldwide, as well as NICAM digital audio standard. It supports most formats for multi-standard TV sound systems such as M/N, B/G, I, L, and D/K. It also provides audio enhancement features, such as AVC, Volume control, Treble/Bass control, Loudness control, Pink noise, Sound Effect, Dolby Pro-Logic, SRS Surround, etc. In addition, the device supports Satellite and FM radio using a TV tuner that includes built-in RF, Local OSC, Converter, and the first Video IF/Sound IF function to receive Satellite and FM radio. Apart from this, it also covers the digital audio formats consisting of I2S, SPDIF, and HDMI audio interfaces.

The audio section consists of automatic sound standard detection; further pilot levels and identification signals can be evaluated internally with subsequent switching between Mono/Stereo and bilingual selection.

The figure below shows the block diagram of the Audio Input Processing block.

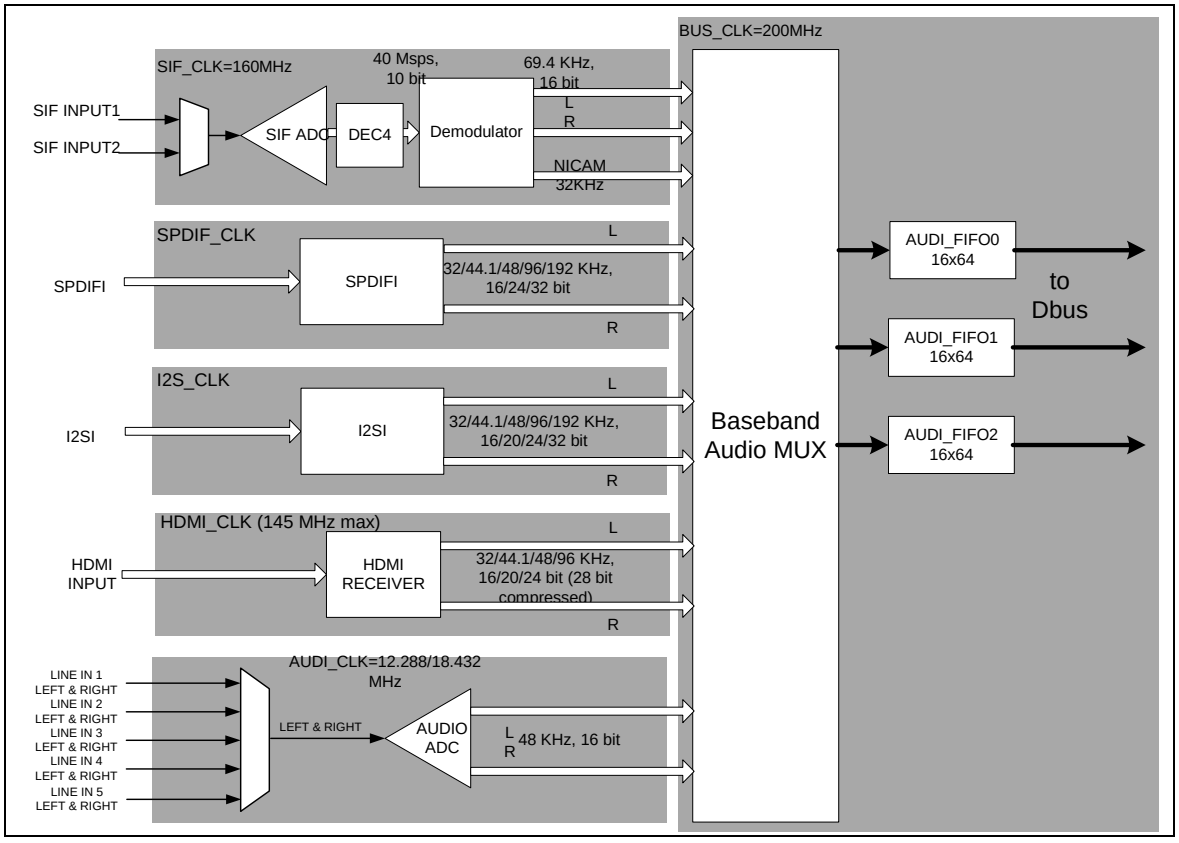


Figure 38. Detailed Audio Input Processing Block Diagram

The Audio Input Processing block consists of standard-definition SIF demodulator, HDMI audio decoder, I2S receiver, SPDIF receiver, and two baseband audio ADCs. Each of these blocks generates audio output with a varying sampling rate and sample size. The Audio Input Processing block then selects the audio output from these blocks and sends the audio data over the 64-bit DMA bus to the DRAM audio buffers. There are three DMA channels in the Audio Input Processing block. The audio buffers are then accessed by the audio DSP (MPE) for further audio processing.

The list below provides an overview of TV standards that are supported by FLI106xx:

Table 32. TV Standards

Country	Video Standard	Audio Standard	SC- Freq	
			CH1	CH2
Argentina	PAL-N	FM-Mono	4.5	
		BTSC	4.5	
Brazil	PAL-M	FM-Mono	4.5	
		BTSC	4.5	
Centr.Africal.Repl	SECAM D/K/K1	FM-Mono	6.5	
		FM-A2	6.5	6.26
		NICAM DK	6.5	5.85
		NICAM DC	6.5	6.88
China	PAL-D/K	FM-Mono	6.5	
		FM-A2	6.5	6.25
		NICAM DK	6.5	5.85
		NICAM DC	6.5	6.88
Czech Republic	PAL-D/K	FM-Mono	6.5	
		FM-A2	6.5	6.26
		NICAM DK	6.5	5.85
		NICAM DC	6.5	6.88
France	SECAM-L	AM-L	6.5	
		NICAM-L	6.5	5.85
Germany	PAL-B/G	FM-Mono	5.5	
		FM-Germany	5.5	5.74
		NICAM-B/G	5.5	5.85
Greece	SECAM-B/G	FM-Mono	5.5	
		FM-Germany	5.5	5.74
		NICAM-B/G	5.5	5.85
Hungary	SECAM D/K/K1	FM-Mono	6.5	
		FM-A2	6.5	6.26
		NICAM DK	6.5	5.85
		NICAM DC	6.5	6.88
Japan	NTSC-M	FM-Mono	4.5	
		FM-Korea	4.5	4.72
		BTSC	4.5	
Korea	NTSC-M	FM-Mono	4.5	
		FM-Korea	4.5	4.72
		BTSC	4.5	
Mexico	NTSC-M	FM-Mono	4.5	
		FM-Korea	4.5	4.72
		BTSC	4.5	
Poland	SECAM D/K/K1	FM-Mono	6.5	
		FM-A2	6.5	6.26
		NICAM DK	6.5	5.85
		NICAM DC	6.5	6.88

Country	Video Standard	Audio Standard	SC- Freq	
			CH1	CH2
Romania	PAL-D/K	FM-Mono	6.5	
		FM-A2	6.5	6.25
		NICAM DK	6.5	5.85
		NICAM DC	6.5	6.88
Russian Fed	SECAM D/K/K1	FM-Mono	6.5	
		FM-A2	6.5	6.26
		NICAM DK	6.5	5.85
		NICAM DC	6.5	6.88
Slovakia	PAL-B/G	FM-Mono	5.5	
		FM-Germany	5.5	5.74
		NICAM-B/G	5.5	5.85
Spain	PAL-B/G	FM-Mono	5.5	
		FM-Germany	5.5	5.74
		NICAM-B/G	5.5	5.85
Sweden	PAL-B/G	FM-Mono	5.5	
		FM-Germany	5.5	5.74
		NICAM-B/G	5.5	5.85
United Kingdom	PAL-I	FM-Mono	6	
		NICAM-I	6	6.55
USA	NTSC-M	FM-Mono	4.5	
		FM-Korea	4.5	4.72
		BTSC	4.5	

17.1 FUNCTIONAL DESCRIPTION

The "Analog Input Function" is responsible for selecting and capturing the desired Sound IF carrier input or external analog audio signals. To reduce the overall application cost, the device will provide two analog switching capabilities for SIF carrier signals and analog audio signals. Analog audio signals will be from different sources (e.g. DVD, VCR, STB, AUX, and mono audio output of TV tuner) depending on the end application. The table below details the summary of all input signals.

Table 33. Types of Audio Input

Input Port #	Type	Source (Example)	Description
1	SIF carrier 1	Tuner 1	Sound IF carrier-1 signal input to demodulator.
2	SIF carrier 2	Tuner 2	Sound IF carrier-2 signal input to demodulator.
3,4	Stereo Audio L1, R1	DVD	Base band stereo audio signal from DVD Player/ Recorder.
5,6	Stereo Audio L2, R2	VCR	Base band stereo/ mono audio signal from VCR, Camcorder, etc.
7,8	Stereo Audio L3, R3	STB	Base band stereo audio signal from Set-Top Box, HDTV, TV
9,10	Stereo Audio L4, R4	AUX	Base band stereo audio signal from other audio source such as S-video, CD, MP3, PC, etc.
11	Mono Audio 1	Tuner 1	Base band mono audio signal from SECAM-L type tuner.

17.2 ARCHITECTURE

17.2.1 SOUND IF PROCESSING

17.2.1.1 ANALOG SOUND IF INPUT

Two SIF inputs (SIF1 and SIF2) can be connected to the analog video input pins of the AFE and routed using on-chip multiplexers to one of the three on-chip video speed A/D converters. An analog automatic gain circuit allows a wide range of input levels. The high pass filters are formed by using the coupling capacitors at the SIF pins.

The Sound IF carrier signals may originate from the SIF output port of the 2-in-1 tuner, such as the Philips FQ1216ME/FQ1236MK3, Samsung TCPN9091PC27A, etc. In addition, FLI106xx is able to handle the FM–Radio standard.

17.2.1.2 DEMODULATOR: STANDARDS AND FEATURES

FLI106xx supports demodulation of all TV Standards worldwide, including the digital NICAM system.

- **A2 Systems**: Detection and demodulation of two separate FM Carriers (FM1 and FM2); demodulation and evaluation of the identification signal of carrier FM2.
- **NICAM Systems**: Demodulation and decoding of the NICAM carrier, detection and demodulation of the analog (AM or FM) carrier. For D/K NICAM, the FM carrier may have a maximum deviation of TBD Hz.
- **BTSC Stereo**: Detection and FM demodulation of the aural carrier resulting in the MTS/MPX signal. Detection and evaluation of the pilot carrier, AM demodulation of the (L-R) carrier, and detection of the SAP subcarrier.
- **BTSC-Mono and SAP**: Detection and FM demodulation of the aural carrier resulting in the MTS/MPX signal. Detection and evaluation of the pilot carrier, detection and FM demodulation of the SAP subcarrier.
- **Japan Stereo**: Detection and FM demodulation of the aural carrier resulting in the MPX signal, Demodulation and evaluation of the identification signal and FM demodulation of the (L-R) carrier.
- **FM Satellite Sound**: Demodulation of one or two FM carriers. Processing of High deviation mono or narrow bandwidth mono, stereo or bilingual satellite sound according to the ASTRA specification.

The key features of the audio blocks are:

- **Standard detection:** The controlling of the demodulator is minimized. All parameters, such as tuning frequencies or filter bandwidth, are adjusted automatically by transmitting one single value to the STANDARD SELECT register.
- **Automatic standard detection:** If the TV sound standard is unknown, the chip automatically detects the actual standard, and then switches to that standard.

17.2.1.3 PREPROCESSING OF DEMODULATOR SIGNALS

The NICAM signals must be processed by the de-emphasis filter and adjusted in level. The analog demodulated signals must be processed by a de-emphasis filter adjusted in level and de-matrixed. The level adjustment must be performed by means of the FM/AM and NICAM pre-scale registers. The necessary de-matrix function depends on the selected sound standard and the actual broadcast sound mode (mono, stereo, or bilingual).

17.2.1.4 SOURCE SELECTION AND OUTPUT CHANNEL MATRIX

The source selector makes it possible to distribute all source signals (one of the demodulator source channels, SCART, or I2S in/out) to the desired output channels (e.g. loudspeaker, headphone, etc.). All input and output signals can be processed simultaneously. Each source channel is identified by selecting the proper register settings.

17.2.2 AUDIO BASEBAND PROCESSING

17.2.2.1 AUTOMATIC VOLUME CORRECTION (AVC)

Different sound sources (e.g. playback systems, VCRs, SCART) will not have the same volume level. Advertisements during movies usually are at a higher volume than the movie itself resulting in annoying volume changes. The automatic volume correction solves this problem by equalizing the volume level.

To prevent clipping, the AVC's gain decreases quickly in dynamic boost conditions. To suppress oscillation effects, the gain increases gradually for low level inputs.

17.2.2.2 VOLUME CONTROL

FLI106xx supports the Volume Control function, which reduces/increases sound levels for the main speaker, subwoofer, and headphones set output. The sound level should be changed from + 12 dB to – 114 dB by 0.125 ~ 0.25 dB step. In addition, it supports three modes of sound level control, such as L-Ch mode, R-Ch mode, and L+R Ch mode.

17.2.2.3 BALANCE CONTROL

FLI106xx supports the Balance Control function, which controls level differences between the Left channel and the Right channel by negative and positive mode. If the level is down compared to the other channel, it is in negative mode; if the level is up, it is in positive mode.

17.2.2.4 BASS CONTROL

FLI106xx supports the Bass Control function, which boosts the up/down output levels of the low frequency of the Left and Right channels for the main speaker/headphones set.

17.2.2.5 TREBLE CONTROL

FLI106xx supports the Treble Control function, which boosts the up/down output levels of the high frequency of the Left and Right channels for the main speaker/headphones set.

17.2.2.6 LOUDNESS CONTROL

FLI106xx supports the Loudness On/Off Control function, which modifies the frequency response and output levels of low and high frequencies for the Left and Right channels under the half volume position only.

17.2.2.7 QUASI-PEAK DETECTOR

FLI106xx supports Quasi-Peak Detection, which adjusts all input audio signals to the same normal listening loudness.

17.2.2.8 SOUND EQUALIZER EFFECT

FLI106xx supports the Sound Equalizer Effect function, which controls frequency response and output levels of five or seven frequencies for the Left and Right channels.

17.2.2.9 LOUDSPEAKER AND HEADPHONE OUTPUTS

The baseband features are implemented in the loudspeaker and headphone output channels. The processing includes bass/treble, loudness, balance, and volume. The loudspeaker output consists of left and right audio out, along with the subwoofer filter output.

17.2.2.10 SUBWOOFER OUTPUT

The subwoofer signal is created by combining the left and right channels directly behind the loudness block using the formula (L+R)/2. Due to the division by 2, the D/A converter is not overloaded, even with full scale input signals. The subwoofer signal is filtered by a third order low pass with programmable corner frequency followed by a level adjustment. At the loud speaker channels, a complementary high-pass filter can be switched on. Subwoofer and loudspeaker output use the same volume.

18 ANALOG VIDEO FRONT END

18.1 ANALOG FRONT END (AFE)

The FLI106xx chip has a sophisticated Analog Front End (AFE) with 16 configurable inputs, in addition to four fixed inputs, input analog MUXes, anti-aliasing filters, and different clamp options for supporting SD and HD video inputs and the Analog to Digital Converters (ADCs). These integrated features eliminate the need for any devices between the input connector and the FLI106xx pin.

The FLI106xx AFE also supports the Video Mixer.

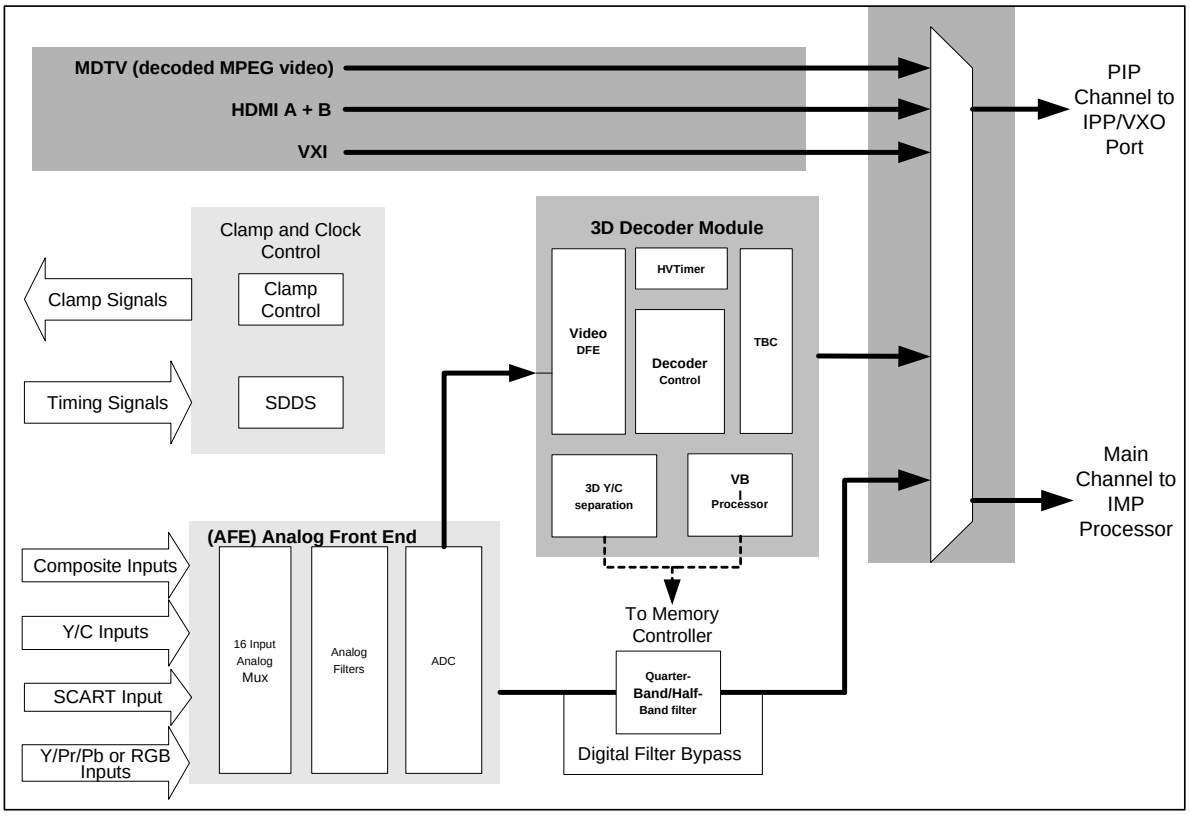


Figure 39. Analog Front End

The figure above depicts the data path for the AFE and Decoder blocks with connections to the input multiplexer. The data path selects whether the data follows the Main Video Channel or PIP video channel.

18.1.1 16-CHANNEL ANALOG INPUT MULTIPLEXER

The Analog Front End provides the capability to capture 16 analog video inputs, which can be a combination of any of the following inputs from the tuner: Composite (CVBS), S-Video (SY, SC), YPrPb (Y, Pr, Pb) or RGB (R, G, B) and SCART and SIF.

18.1.2 INPUT FILTERING

The front end provides filtering capability depending on the type of input video signal in use. The use of these filters eliminates the need to have any external filter components. The filters included are both in the analog, as well as digital domain. These filters help to reduce the anti-aliasing artifacts.

Figure 40 below illustrates the analog and digital filter combinations implemented in the front end.

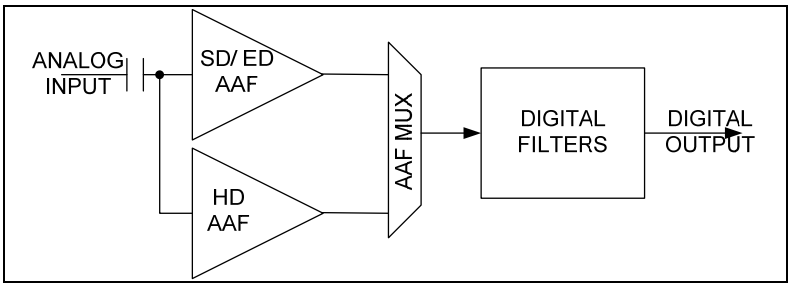


Figure 40. Analog and Digital Filter Combos

Table 34 illustrates the Pass Band frequencies for analog and digital anti-aliasing filters.

Table 34. Pass Band Frequencies for Analog and Digital Anti-Aliasing Filter

	Filter	FS	F1
1	Analog SD_ED AAF	NA	14 MHz
2	Analog HD AAF	NA	37 MHz
3	Digital AAF SD	80 MHz	6.5 MHz
4	Digital AAF ED	108 MHz	13.5 MHz
5	Digital AAF HD	148.5 MHz	29.5 MHz
6	Digital Filter HD 1080P	148.5 MHz	NA

The digital filters are implemented as a combination of HB1, HB2, and LPF filters.

18.1.3 AFE PIN CONNECTION

The necessary external hardware is integrated into the Analog Front End of the FLI106xx to support direct connection to the physical analog signal connector on the PCB. Only an AC coupling capacitor and a termination resistor are needed between the physical connector and the chip.

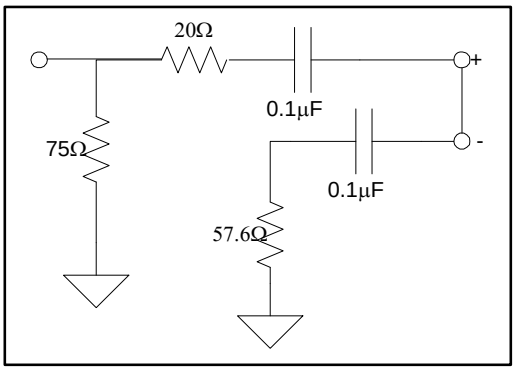


Figure 41. Sample Connection for Analog Input Signal

Note: It is important to follow the recommended layout guidelines for the circuit. These are described in the "Layout Guidelines".

18.1.4 ADC CHARACTERISTICS

The device supports quadruple 10-bit ADCs with one ADC dedicated for SIF and one 6-bit Sync ADC. The table below summarizes the characteristics of the ADC:

Table 35. ADC Characteristics

ADC Characteristics	Min	Typ	Max	Units
Track & Hold Amp Bandwidth	20		400	MHz
Full Scale Adjust Range at AFE Inputs - Without AAF - With AAF			 0.9 1.6	V
Full Scale Adjust Sensitivity		0.0051		
Zero Scale Adjust Sensitivity		3		LSB
Sampling Frequency (Fs)		0-165		MHz
Differential Nonlinearity (DNL)		+/- 0.5		LSB
Integral Nonlinearity (INL)		4.5		LSB

The AFE supports the circuit for calibrating the track and hold amplifiers before the data is fed to the ADC.

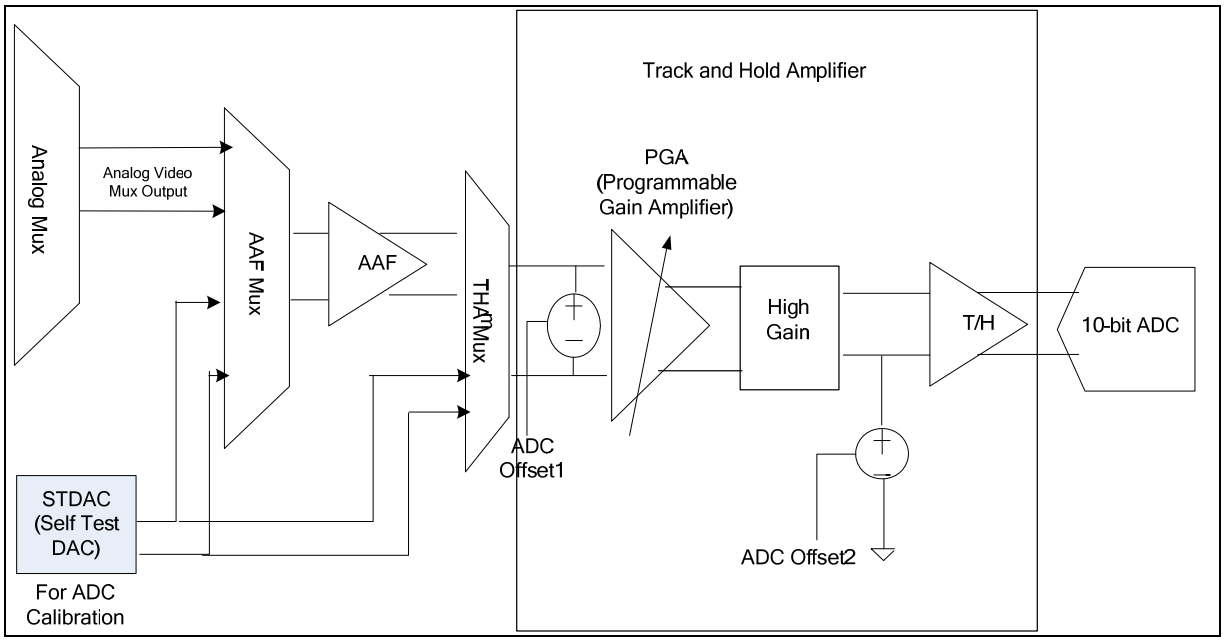


Figure 42. AFE Support for Calibration Circuit

18.1.5 CLAMPING

In order to ensure that the input AC coupled signals are within the dynamic range of the ADC, the device supports different clamp circuitry for different input types.

The input video signals (Graphics, Component, S-Video, or CVBS) are held at the reference voltage irrespective of the APL of the input video by the clamp logic.

18.1.5.1 TYPES OF CLAMPING

Hard Clamping for Graphics and Component inputs:

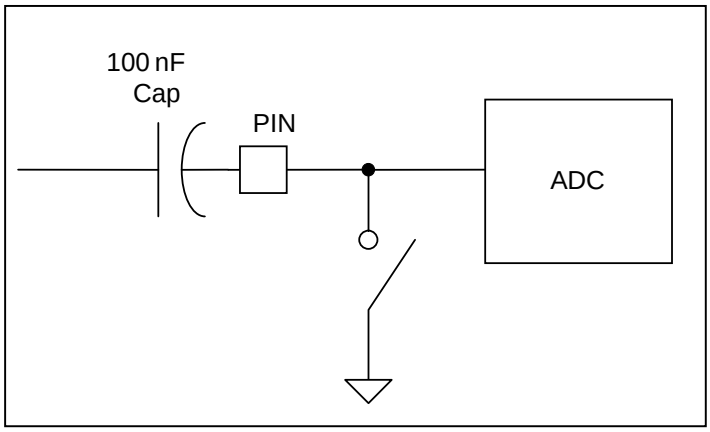


Figure 43. Hard Clamping Circuit

Video Loop Clamping for CVBS, S-video, or RF inputs:

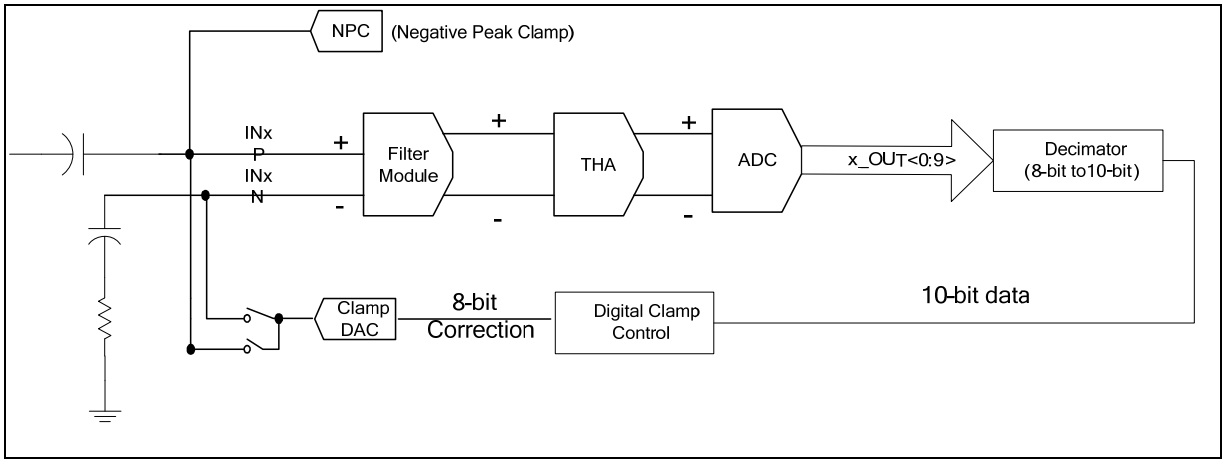


Figure 44. Loop Clamping Circuit

The digital clamp feedback loop control is a closed loop feedback system employed in the device to achieve video clamping. This method of clamping involves a digital feedback control loop, a D-to-A converter acting like a charge pump in conjunction with an external Clamp Capacitor.

18.1.6 CLOCK RECOVERY CIRCUIT

The Source Direct Digital Synthesis (SDDS) clock recovery circuit generates the clock used to sample analog RGB data (IP_CLK or source clock). SDDS is also used for capturing video (up to 150 MHz closed loop for HD, 160 MHz openloop for SD, 135 MHz for VGA). This circuit is locked to the HSYNC of the incoming video signal.

Patented digital clock synthesis technology makes the FLI106xx clock circuits resistant to temperature/voltage drift. Using DDS (Direct Digital Synthesis) technology, the clock recovery circuit can generate any IP_CLK clock frequency within the range of 10 MHz to 135 MHz.

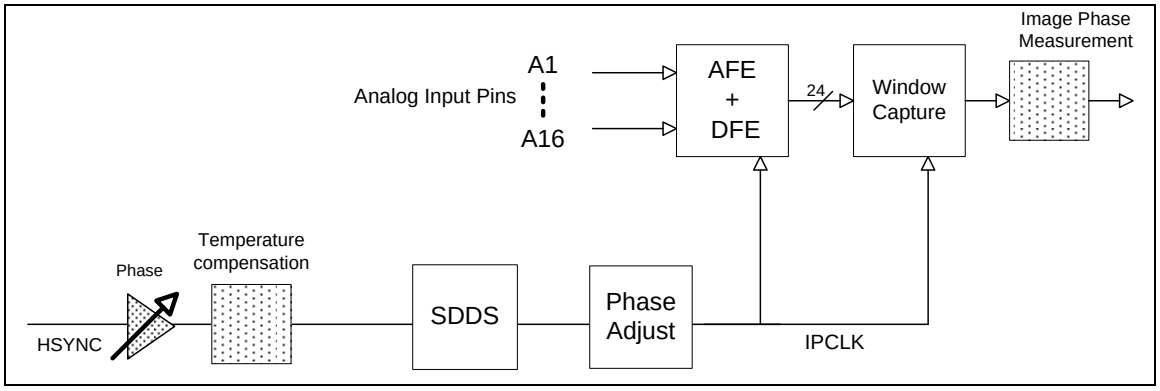


Figure 45. Clock Recovery

18.1.7 SCHMITT TRIGGER

Integrated Schmitt triggers are used for improving the rising and falling edges of the analog Horizontal and Vertical Sync signals. This eliminates the need for using external Schmitt components in the system design. Typical behavior of an integrated Schmitt circuit is shown in the following figure. The integrated Schmitt circuit has 2 sets of threshold voltages for ViH and ViL, which together allows four different hysteresis levels. The Schmitt threshold levels of integrated circuits are in compliance with standard CMOS and TTL input signals. A programmable option is provided in the software for the selection of Schmitt levels.

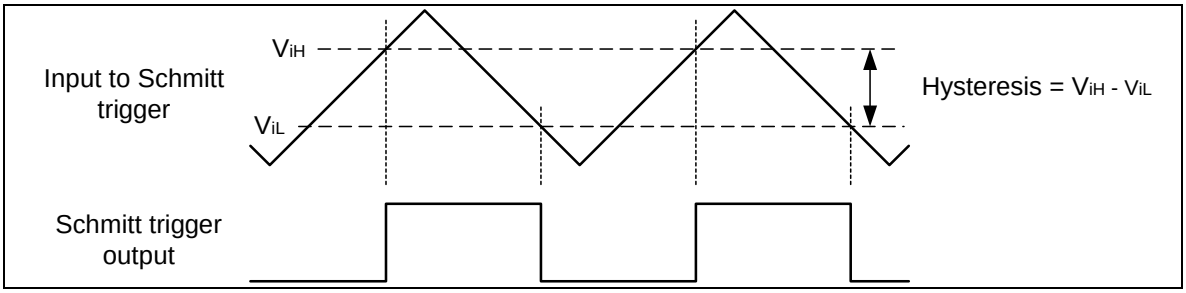


Figure 46. Schmitt Circuit Timing

Table 36. Schmitt Circuit Thresholds

Description	Programmed Threshold Levels
Low to High	Vih[V] = 2.0
High to Low	Vil [V] = 0.8

18.1.8 SAMPLING PHASE ADJUSTMENT

The programmable ADC sampling phase is adjusted by delaying the SDDS clock with respect to the HSYNC input. The accuracy of the sampling phase is checked and the result is read from a register. This feature enables accurate auto-adjustment of the ADC sampling phase.

18.1.9 ADC CAPTURE WINDOW

ADC Capture Window illustrates the capture window used for input. In the horizontal direction, the capture window is defined in IP_CLKs (equivalent to a pixel count). In the vertical direction, it is defined in lines.

All the parameters beginning with "Source" are programmed FLI106xx registers values. The Input Vertical Total is determined solely by the input and is not a programmable parameter.

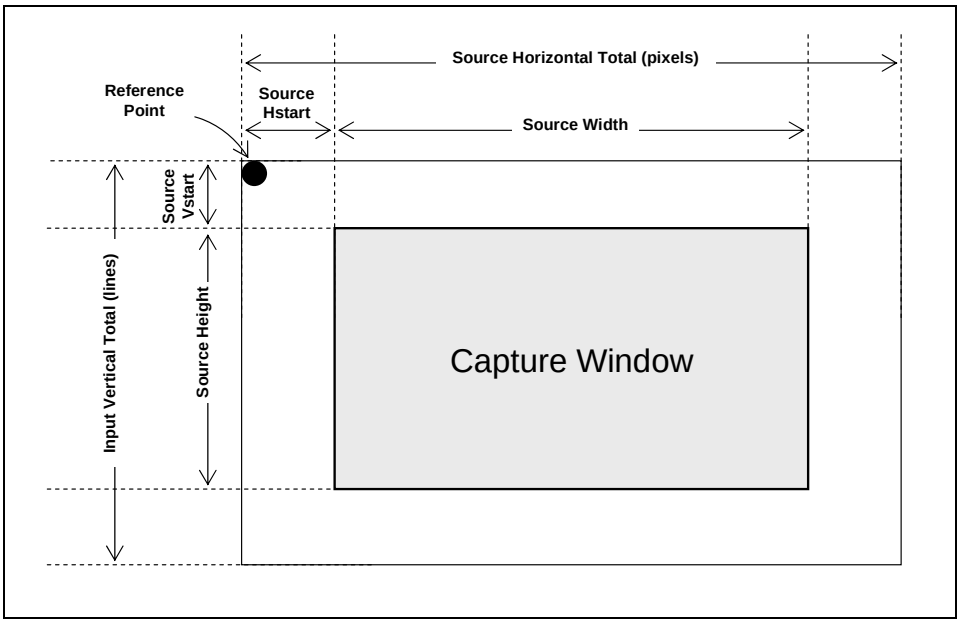


Figure 47. ADC Capture Window

The Reference Point marks the leading edge of the first internal HSYNC following the leading edge of an internal VSYNC. Both the internal HSYNC and the internal VSYNC are derived from external HSYNC and VSYNC inputs.

Horizontal parameters are defined in terms of single-pixel increments relative to the internal horizontal sync. Vertical parameters are defined in terms of single-line increments relative to the internal vertical sync.

For ADC interlaced inputs, the device may be programmed to automatically determine the field type (even or odd) from the VSYNC/HSYNC relative timing. See Section 18.5 Input Format Measurement (IFM) for more information.

18.1.10 DIGITAL PROCESSING AFTER AFE–DIGITAL FRONT END

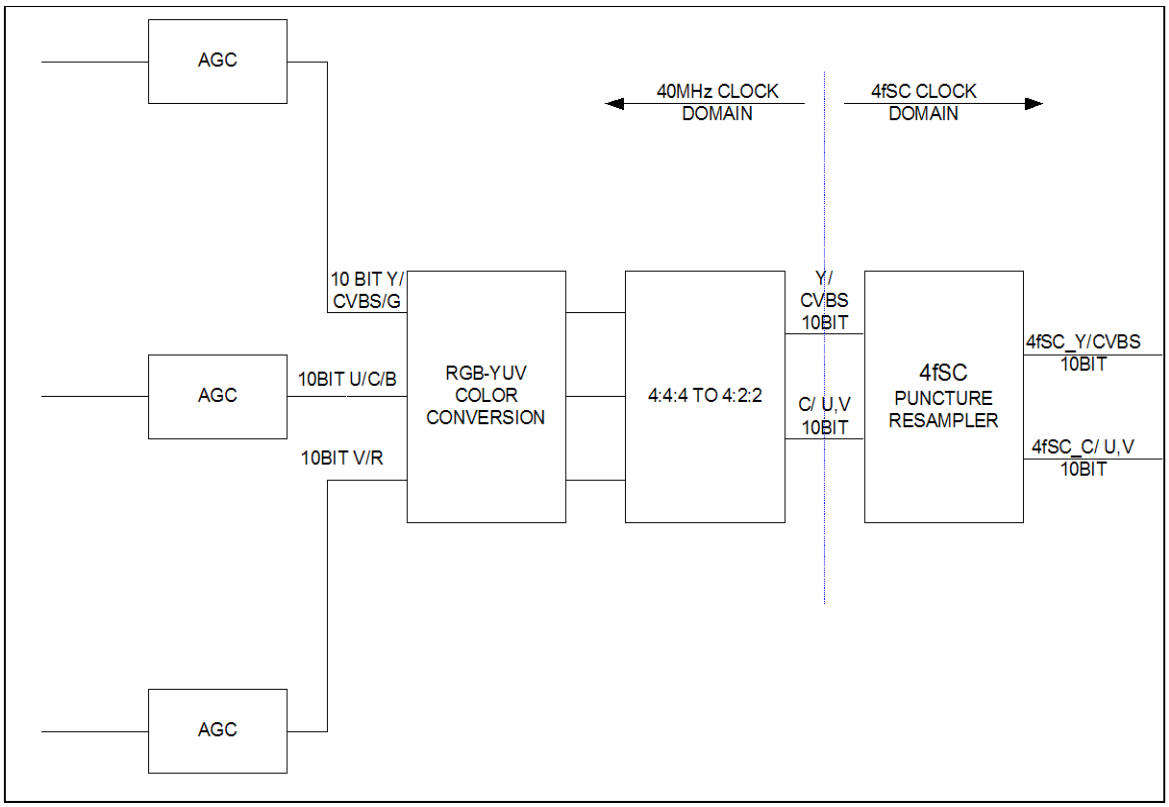


Figure 48. DFE Datapath

The DFE Datapath is shown above. The DFE consists of three channels that can support the following fixed-position formats: Channels 1, 2, and 3 can be either G,B,R or Y,U,V or two channels of Y and C or one channel of CVBS. The DFE performs Digital Clamp Loop Control for each channel, AGC Control, Color Conversion, Chroma Downscaling and 4fSC re-sampling. The input to the DFE is 10-bit 40MHz Data. The output is 4fSC Sampled CVBS, Y, C, or YUV or just 10-bit CVBS.

18.1.11 AUTOMATIC GAIN CONTROL (AGC)

The Analog Front End can provide Automatic Gain Control for video signals that are not within the standard specification range for the input voltage. This feature can be enabled or disabled as necessary by firmware. The AGC compensation can be handled by using a combination of firmware and hardware AGC blocks. The operation characteristic parameters for AGC include:

- AGC gain range (see below):

Input Format	AGC Range (in dB)
Graphics/Component	-2.9 to +2.10
CVBS/Y	-4.55 to +3.75

- Input signal level (AGC pull-in and hold range): 600mV ~ 1.6V.

18.1.12 DIRECT SCART INPUT SUPPORT

The FLI106xx supports direct connection to SCART devices such as DVD players and set-top boxes in Europe by enabling two features in the Analog Front End:

- **Fast Blank Switching:** The device supports Fast Blank switching for SCART connectors or external OSD controllers that require the merging of a line-locked analog RGB stream with the viewed composite or S-Video input. AFE supports Quad Track and Hold block for sampling data in Time Division Multiplexed format. The device supports Static and Dynamic modes of operation for Fast Blank.
- **Composite Output:** The device supports a composite video output signal to enable the connection of external OSD controllers and support for SCART connectors. FLI106xx can support SCART Overlay and SCART Component Modes of Operation.

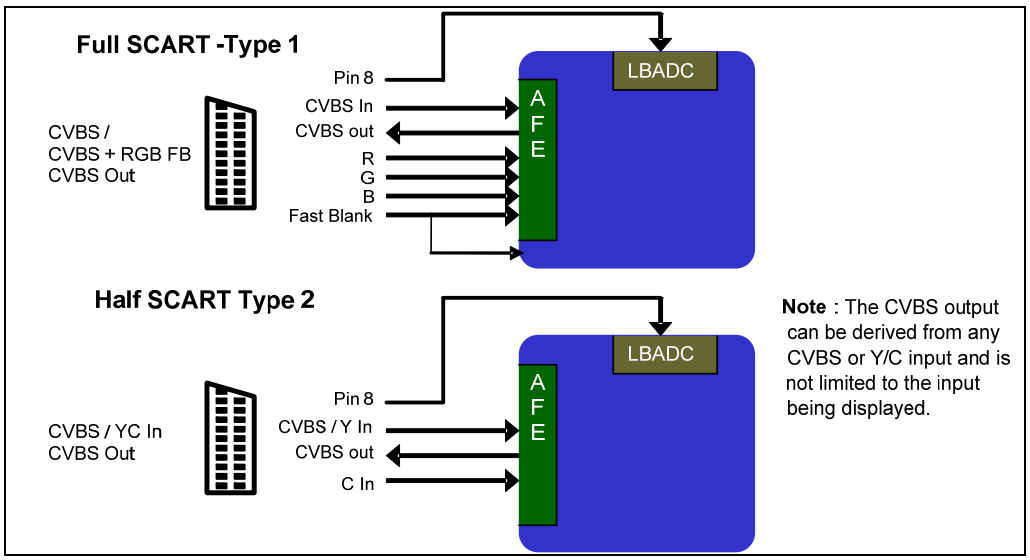


Figure 49. SCART Connectors

18.2 MULTI-STANDARD VIDEO DECODER

The FLI106xx contains an embedded Multi-Standard Video Decoder as can be seen in Figure 50 below:

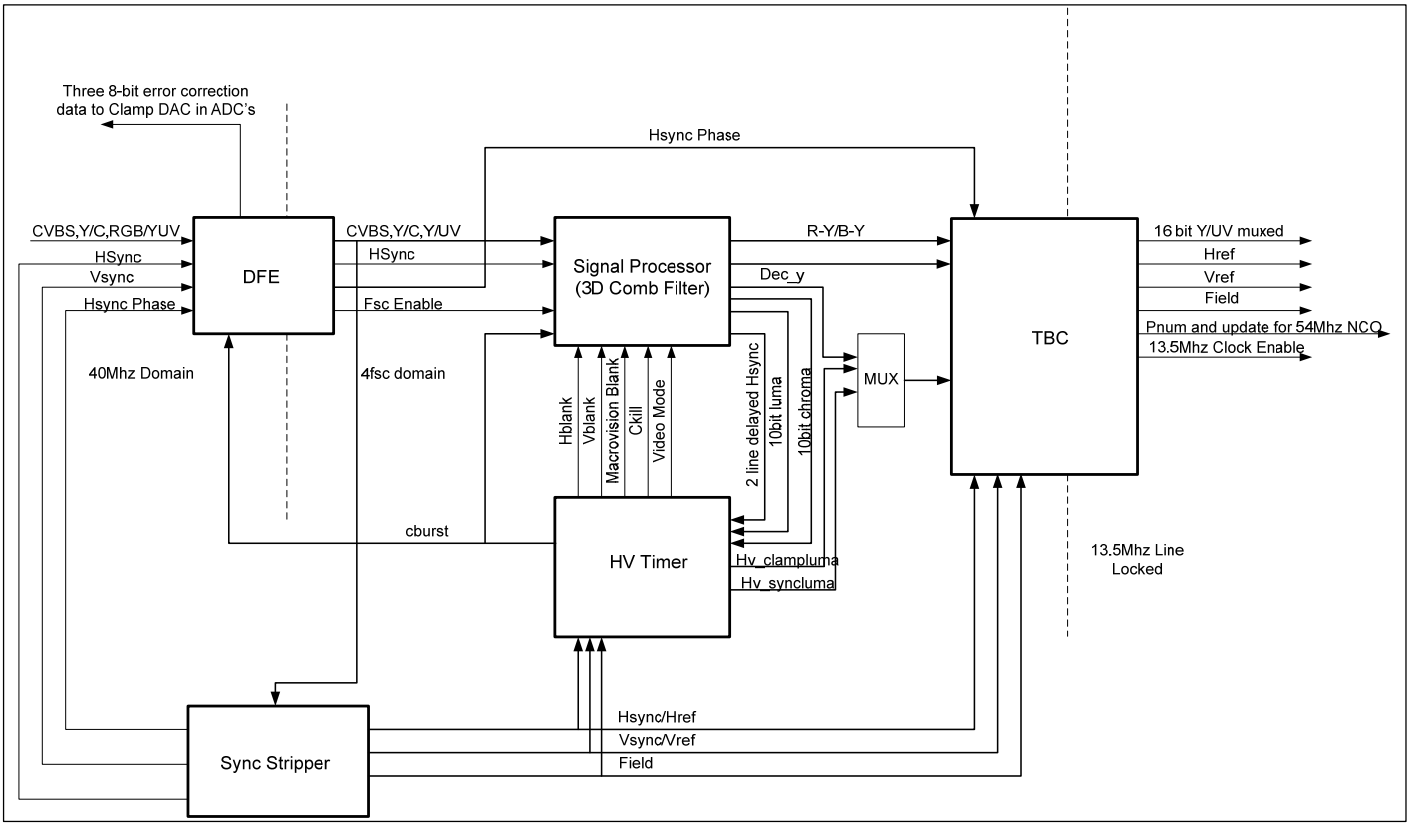


Figure 50. Multi-Standard Video Decoder

This decoder contains a 3D Comb filter for optimum Y/C separation. This reduces or eliminates any artifacts due to poor Y/C separation seen on standard 2D video decoders on Composite inputs, most commonly found on TV tuners.

The decoder supports a Digital Sync Stripper for generating reference sync signals for the signal processing in the decoder. Additionally, the decoder supports a HV timer sub-block for generating mode detection timing flag and supports mode detection state-machine for SD video inputs. The decoder uses a Time Base Correction (TBC) module to stabilize input timing information of the input signal. This will correct unstable or jittery input signals, such as VCR signals or non-standard terrestrial signals, ensuring they are processed with proper timing.

Table 37 summarizes the supported formats and parameters associated with each standard.

Table 37. Supported Formats

Standard	Comb Filtering	Fsc (MHz)	Lines per Field (Hz)
NTSCM	3D Adaptive	3.579545	525 / 59.94
PALI, PALD	3D Adaptive	4.433618	625 / 50
PALM	3D Adaptive	3.5756	525 / 59.94
PALN	3D Adaptive	3.5820	625 / 50
SECAM	2D Adaptive	FM Modulation	625 / 50

Standard	Comb Filtering	Fsc (MHz)	Lines per Field (Hz)
NTSC443	2D Adaptive	4.433618	525 / 59.94
PAL60 (Quasi PAL)	2D Adaptive	4.433618	525 (625 in 525)

Note: F_{sc} is the frequency of the sub-carrier.

The device can auto-detect each of these standards using measurement parameters given by the internal decoder in combination with the Input Format Measurement block. External memory is used for Composite NTSC-M and PAL signal inputs. External memory enables a comparison of the current and previous frame (3D) as well as between lines (2D) to ensure maximum Y/C separation.

18.2.1 COLOR DECODER

The diagram below depicts the color decoding being implemented for FLI106xx. The line memory and the comb logic are used to separate Y and C signals from the CVBS input. The separated Chroma signal is passed through the Chroma AGC, the Chroma Demodulator, and the PAL Averager before generating the U and V signal. All this processing is done at 4fsc domain.

The Y and U/V signals are multiplexed with the Y and U/V outputs from the SECAM Processor before being fed to the TBC in the video decoder.

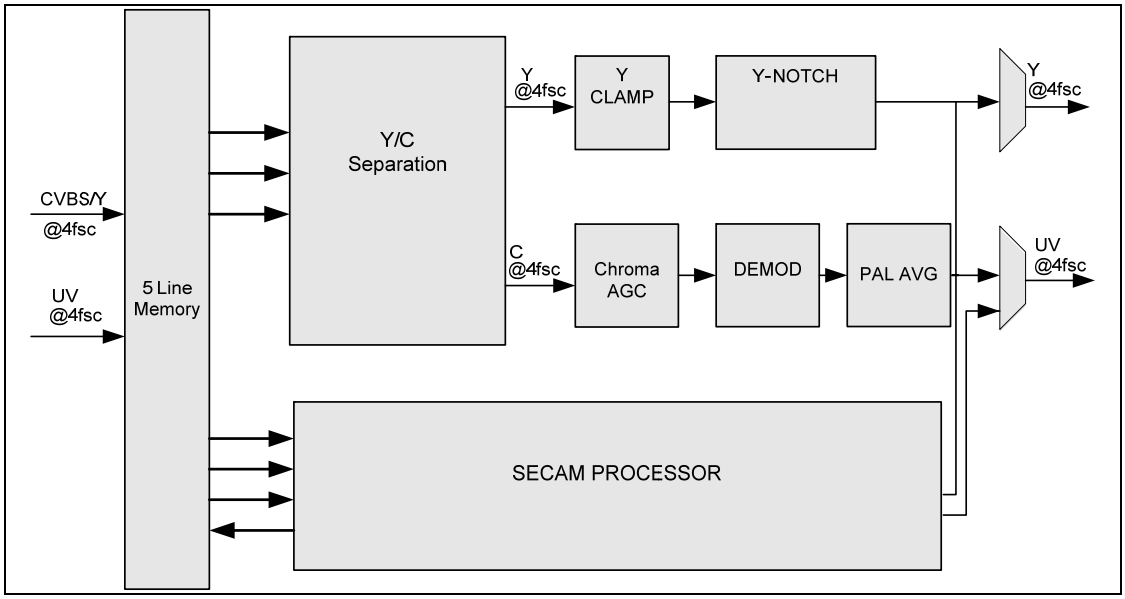


Figure 51. Color Decoding

18.2.2 Y / C SEPARATION

The 2D Y/C Separation module consists of Line memories, Comb Control logic consisting of V-Logic, Single-Line Logic, 2D Adaptive Comb Filter, and the H-Logic. The adaptive control in the Y/C separation block helps to reduce dot crawl and hanging dot artifacts.

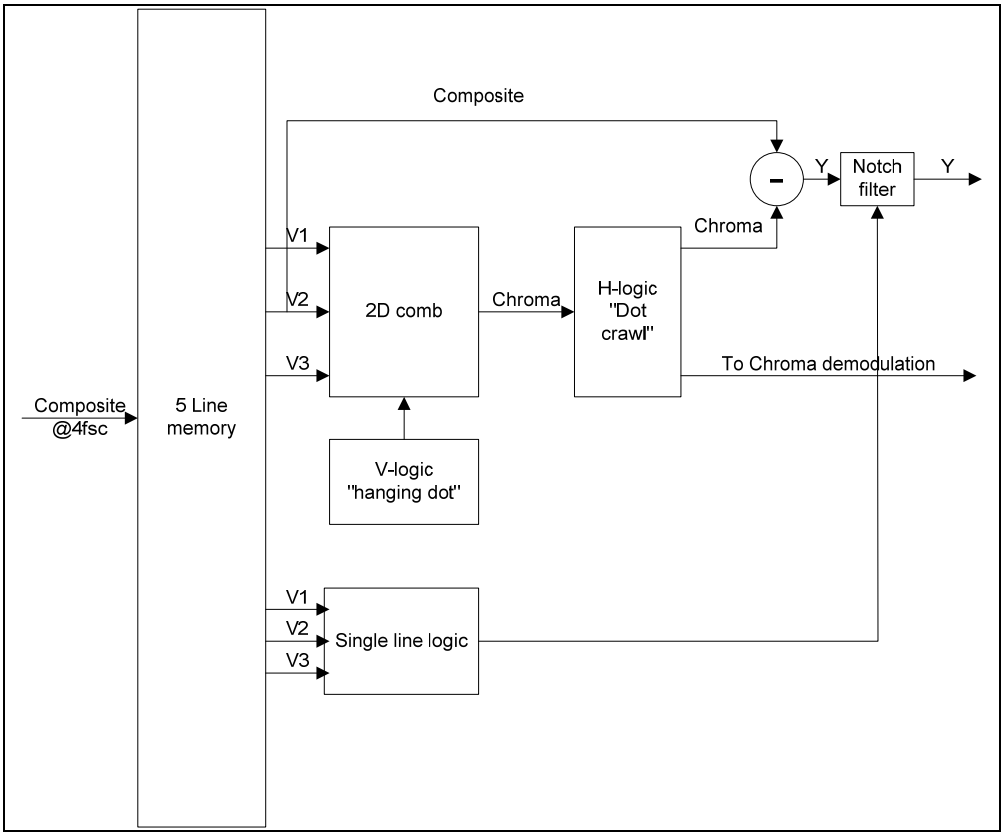


Figure 52. 2D Y/C Separation

The 3D Logic is added on top of adaptive 2D decoder with V-logic, H-logic, and Single-line logic. Chroma signal is the blender output of adaptive 2D and 3D comb filtering based on the motion vector value provided by the motion detection block.

18.2.3 COLOR KILLER OPERATION

The Color Killer converts a color image into a black and white image when the input color signal level is too low or too noisy. The Chroma signal that comes from the Y/C separator is rectified, and then the C burst pulse samples the Chroma signal.

For NTSC/PAL inputs, burst level will be compared with a preset threshold, which could be changed through the register (the output is 0 for burst levels higher than the threshold). For SECAM input signals, color kill is asserted by disabling the RY/BY coefficients output of the Chroma demodulator.

The color killer amplitude levels have programmable hysteresis.

18.2.4 IF COMPENSATION

FLI106xx supports IF Compensation to compensate for the high frequency Chroma content attenuation by off-air or mistuned input signals. The frequency response for the compensation filter is shown in Figure 53.

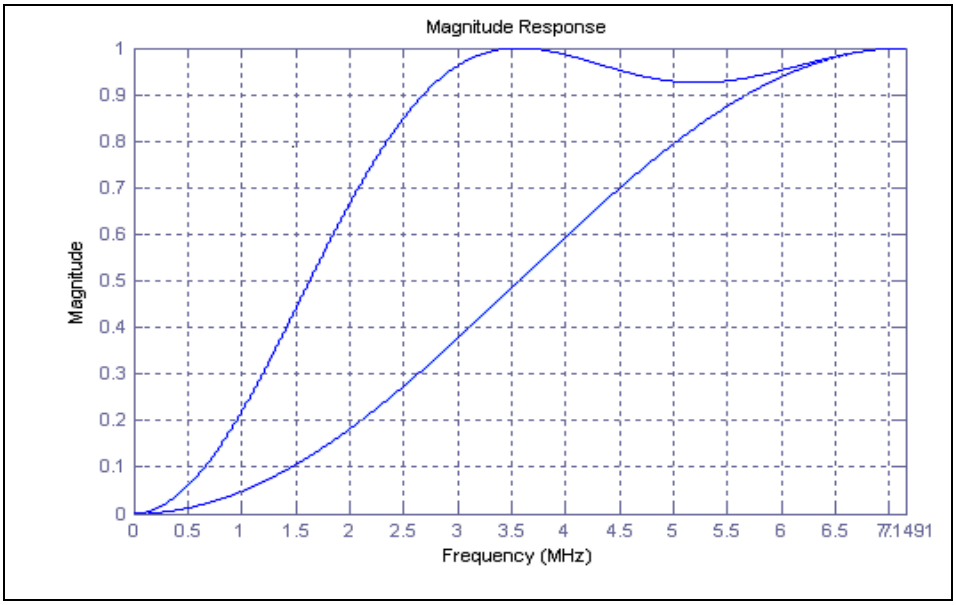


Figure 53. Frequency Response for Compensation Filter

The lower curve represents the filter response for a blend value of 3'b000 and the upper curve represents the filter response for a blend value of 3'b111. The blend and chroma gain and the Luma-Chroma delay controls are programmable by register.

18.2.5 MACROVISION INPUT SUPPORT

The video decoder is capable of accepting video inputs protected by Macrovision™ copyright protection for display on television devices.

18.2.6 ADAPTIVE LUMA PEAKING

The Adaptive Luma Peaking module works on the 2D decoded Luma, which has reduced sharpness in the chroma sub carrier band. To correct this, and to match the 2D data sharpness to near 3D levels, this module boosts the Luma frequency component around chroma sub carrier frequency.

18.2.7 DECODER CHROMA TRANSIENT IMPROVEMENT (CTI)

The CTI module works the demodulated chroma. It increases the rise time between color transitions, without introducing ringing to provide a sharper image.

18.3 HIGH-DEFINITION MULTIMEDIA INTERFACE (HDMI) INPUT

18.3.1 OVERVIEW

HDMI is an acronym for the High-Definition Multimedia Interface—an industry-supported, uncompressed, all digital audio/video interface. HDMI provides an interface between any compatible digital audio/video source, such as a set-top box, DVD player, and A/V receiver and a compatible digital audio and/or video monitor, such as a digital television (DTV).

HDMI can carry high-quality multi-channel audio data and all standard and high-definition consumer electronics video formats. It can also carry control and status information in both directions. Content protection technology is available with HDMI.

The HDMI cable and connectors carry four differential pairs that make up the TMDS data and clock channels. These channels are used to carry video, audio, and auxiliary data. In

addition, HDMI carries a VESA DDC channel. The DDC is used for configuration and status exchange between a single Source and a single Sink. The optional CEC protocol provides high-level control functions between all of the various audiovisual products in a user's environment.

Audio, video, and auxiliary data is transmitted across the three TMDS data channels. The video pixel clock is transmitted on the TMDS clock channel and is used by the receiver as a frequency reference for data recovery on the three TMDS data channels.

HDMI allows a wide variety of explicitly defined video format timings to be transmitted and displayed. These video format timings define the pixel and line counts, and timing, synchronization pulse position, and duration, and determine the format type as interlaced or progressive.

The HDMI Source determines the pixel encoding and video format of the transmitted signal based on the characteristics of the source video, the format and pixel encoding conversions possible at the Source, and the format and pixel encoding capabilities and preferences of the Sink.

It also supports down mixing output for multi-channel inputs such as 5.1 or 7.1 channel input.

18.3.2 HDMI 1.3A RECEIVER

FLI106xx includes two HDMI input interfaces supporting all input modes as described in the HDMI 1.1 and HDMI 1.3 standards (including CEC support) and is backward compatible with the DVI 1.0 specification.

Note: Only one of the HDMI inputs will be displayed at any instance on Main or PIP. Accordingly, the HPD signal of the inactive HDMI port should be negated.

FLI106xx is capable of receiving and outputting two to eight channels of audio output at 192 kHz. Four industry standard I2S outputs provide direct connection to low-cost audio DACs. Additionally, an SPDIF output is available and supports up to 96 kHz audio.

Two and Eight-Channel LPCM (packet type 2), Non-LPCM Compressed audio (packet type 2), One Bit audio (packet type 7), DST (packet type 8), and High Bit-Rate audio (packet type 9) are all routed to the audio decoder via proper audio links.

FLI106xx supports high bandwidth digital content protection (HDCP) 1.3 that is associated with the HDMI interface.

Note: Genesis will provide customers with HDCP keys.

HDMI-related features of the FLI106xx include:

- Expanded color gamut xvYCC, IEC 61966-2-4.
- Deep Color allows for 8, 10, 12, and 16-bit color depth
- VESA input modes up to WQXGA resolution
- Full range of RGB quantization
- Supports basic (IEC-60958) and compressed audio (IEC-61937)
- Supports all formats specified in CEA 861-D
- Supports Active Format Description (ETSI TS 101 154 v1.7.1, Digital Video Broadcasting)
- Supports E-EDID 1.3 specification

- Supports two categories of cables: Cat 1 (up to 74.25MHz) and Cat 2 (between 74.25MHz and 225MHz)

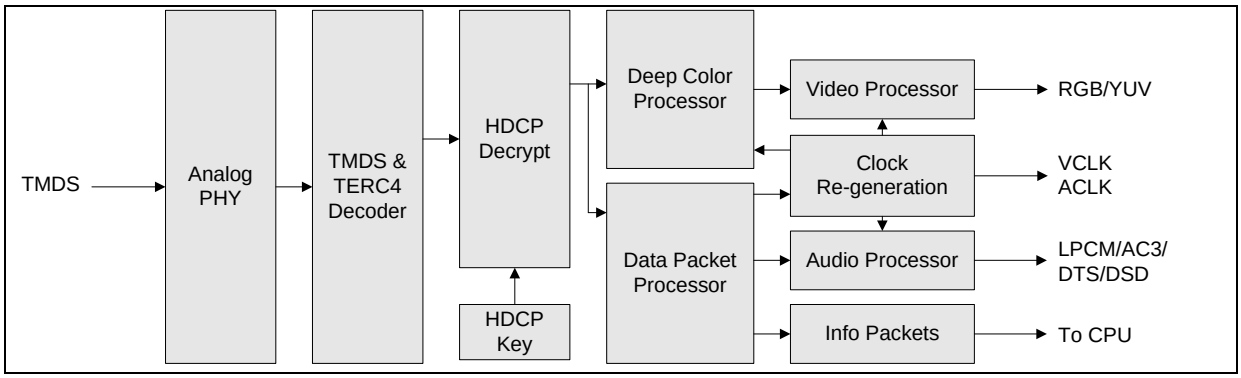


Figure 54. HDMI 1.3a Receiver Block

The components of the HDMI 1.3a receiver block on the FLI106xx are shown above. Main components include:

- **Analog PHY:** designed to support operating speeds of up to 225MHz.
- **TMDS and TERC4 Decoder:** converts the 10 bits per channel data into data bytes.
- **HDCP Decrypt Module:** converts content protected data back to raw data.
- **Data Packet Processor:** de-muxes audio and information packets from the source and sends this to the CPU and Audio Processor Module.
- **Deep Color Processor:** converts byte-packed 10-bit and 12-bit video to the video stream.
- **Video Processor Module:** provides video mute processing and dither support for the internal 10bpc video path.

18.3.3 AUDIO PROCESSING LOGIC

The audio processing logic receives the audio stream packets from the HDMI data bus and puts them into an audio FIFO. The audio sample clock is recovered using information in the N/CTS packets and uses that sample clock to pull the data out of the FIFO. The audio data can be outputted from the receiver as either SPDIF or I2S or both.

Audio sample rates from 32KHz to 192KHz are supported by the I2S outputs and 32-96KHz by the SPDIF output. SPDIF output supports both L-PCM and compressed audio formats including AC3. See the table below for supported audio formats.

Table 38. Audio Formats Supported

Layout Value	Original Audio	Sample Fre KHz	Bits/Sample
Layout 0	L-PCM 2 Ch, AC3	32, 44.1, 48, 96, 192	16, 20, 24

18.4 MULTI-STANDARD VBI DATA PROCESSING

The FLI106xx microprocessor can extract a wide range of Vertical Blanking Interval (VBI) data and display this data with the 2D Blit Engine. The microprocessor can support VBI data extraction in the Main channel, as well as in the background when the corresponding video stream is not present on the output stream.

18.4.1 VBI SLICER

The VBI data slicer can extract data encoded into the vertical blanking interval of the input video stream. The VBI data slicer can support the following standards:

- Closed Caption Standards:
  - EIA-608: US Closed Caption standard
  - EIA-608: EDS (including CGMS-A, V-Chip) standard
- WSS Standards:
  - BT.1119: WSS Information for NTSC and PAL video signals
  - IEC 61880: WSS Information for NTSC video signals
  - EIA-J CPR-1204: WSS Information for NTSC video signals
  - ETSI EN 300 294: WSS Information for PAL video signals
- Teletext Standards:
  - BT 653: World Standard Teletext (System A, B, C & D)
  - EIA-516: North American Broadcast Teletext Specification (NABTS)
  - ETSI ETS 300 706: PAL Teletext Standard
- Program Delivery Control (PDC), VPS, Electronic Program Guide (EPG) & Program System Information Protocol (PSIP):
  - ETSI ETS 300 707: EPG standard for PAL
  - ETSC ETS 300 231: PDC for PAL standard
  - Gemstar
- Copy Generation Management System (CGMS) information carried on Component Video Interface (CVI):
  - EIA/CEA-805: Data Service on the Component Video Interfaces

18.4.2 TTX LEVELS

Presentation Level 1.0:

- Defines basic Teletext page: 24 Rows of 40 Characters
- Alpha Mosaic Characters
- Fixed Color Palette
- Spacing Attributes

Presentation Level 1.5:

- Extends the character repertoire via packets X/26

Presentation Level 2.5:

- Extend the language repertoire
- Increased color palette: 16 of the 32 being re-definable
- Allows a number of simple re-definable characters

18.4.3 VBI DATA PROCESSING

VBI decoder consists of one high quality equalizer. This equalizer is designed to eliminate or reduce ISI. Decoded data bytes from the VBI data slicer are stored in a memory buffer for subsequent processing by the microprocessor. The microprocessor decodes the different formats of the VBI data and passes the information to the OSD controller for output to the display controller.

The VBI decoding process can run continuously in the background for inputs that contain VBI data. This will be for the purposes of CC, Fast Page (instant retrieval of Teletext pages), WSS, and EPG style services. For Fast Page functionality, storage support of up to 4096 decoded Teletext pages is available.

18.5 INPUT FORMAT MEASUREMENT (IFM)

The FLI106xx supports three Input Format Measurement (IFM) blocks. These blocks can be independently used for measuring the horizontal and vertical timings of Main and PIP inputs. The third IFM can be used for detecting the change in either the Main or PIP input to configure the system accordingly.

The IFM features a programmable reset, separate from the regular FLI106xx soft reset. This reset disables the IFM, reducing power consumption. The IFM is capable of operating while the FLI106xx is running in power-down mode.

Horizontal measurements are assessed in terms of the selected IFM_CLK—either TCLK (normal operation) or OCM_CLK/2—while vertical measurements are assessed in terms of HSync pulses.

For an overview of the internally synthesized clocks, see Section 7. Clock Generation.

18.5.1 HSYNC / VSYNC DELAY

The active input region captured by the FLI106xx is specified with respect to internal HSync and VSync. By default, internal syncs are equivalent to the HSync and VSync at the input pins and thus force the captured region to be bounded by external HSync and VSync timing. However, the FLI106xx provides an internal HSync and VSync delay feature that removes this limitation. By delaying the sync internally, the FLI106xx can capture data that spans across the sync pulse.

It is possible to use HSync and VSync delay for image positioning. (Alternatively, Source_HSTART and Source_VSTART in Figure 47 ADC Capture Window are used for image positioning of analog input). The intentional movement of images across apparent HSync and VSync boundaries creates a horizontal and/or vertical wrap effect.

HSYNC is delayed by a programmed number of selected input clocks.

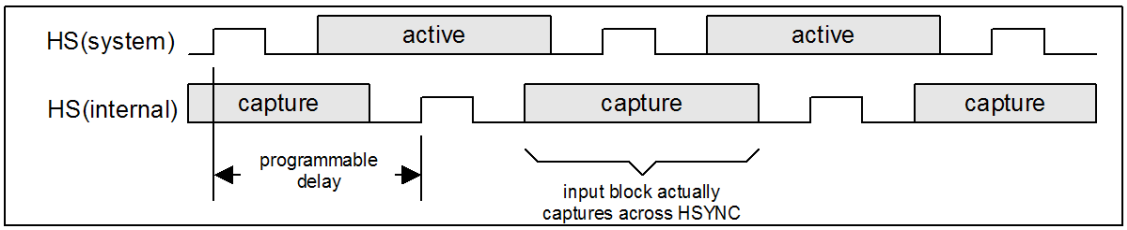


Figure 55. HSync Delay

Delayed horizontal sync may be used to solve a problem with VSync jitter with respect to HSync. VSync and HSync are generally driven active with different paths to the FLI106xx (HSync is often regenerated from a PLL). As a result, VSync may be seen earlier or later. Because VSync is used to reset the line counter and HSync is used to increment it, any difference in the relative position of HSync and VSync is seen on-screen as vertical jitter. By delaying the HSync a small amount, it can be ensured that VSync always resets the line counter prior to it being incremented by the "first" HSync.

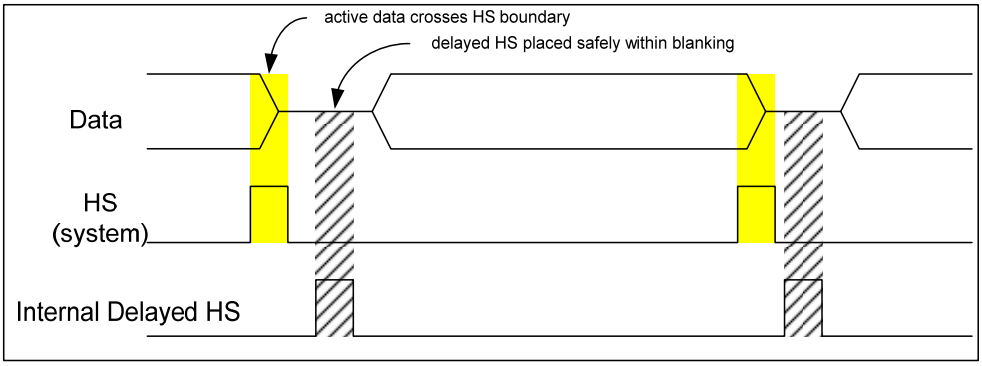


Figure 56. Active Data Crosses HSYNC Boundary

18.5.2 HORIZONTAL AND VERTICAL MEASUREMENT

The IFM is able to measure the horizontal period and active high pulse width of the HSync signal, in terms of the selected clock period (either TCLK or OCM_CLK/2). Horizontal measurements are performed only on a single line per frame (or field). The line used is programmable. It measures the vertical period and VSync pulse width in terms of rising edges of HSync.

Once enabled, measurement begins on the rising VSync and is completed on the following rising VSync. Measurements are made on every field/frame until disabled.

18.5.3 FORMAT CHANGE DETECTION

The IFM is able to detect changes in the input format relative to the last measurement and then alerts both the system and the on-chip microcontroller. The microcontroller sets a measurement difference threshold separately for horizontal and vertical timing. If the current field/frame timing is different from the previously captured measurement by an amount exceeding this threshold, a status bit is set. An interrupt can also be programmed to occur.

18.5.4 WATCHDOG

The watchdog monitors input VSync/HSync. When any HSync/VSync period exceeds the programmed timing threshold, status bits are set. An interrupt can also be programmed to occur.

18.5.5 INTERNAL ODD / EVEN FIELD DETECTION

The IFM has the ability to perform field decoding of interlaced inputs to the ADC via two methods.

The first method consists of internal hardware counting the number of lines between v-sync pulses for the current field. If the count is found to be an "odd" number, then the next field is marked "even". This first method is recommended to be used.

The second method consists of the user specifying start and end values to outline a "window" relative to HSync. If the VSync leading edge occurs within this window, the IFM signals the start of an ODD field. If the VSync leading edge occurs outside this window, an EVEN field is indicated (the interpretation of odd and even can be reversed). The window start and end points are selected from a predefined set of values.

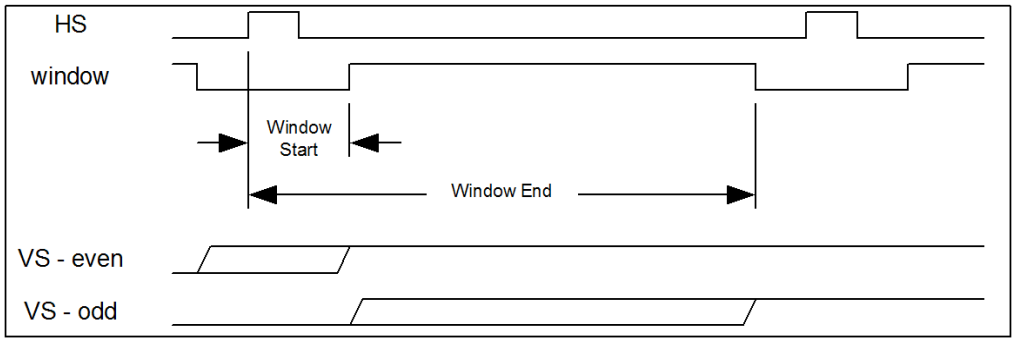


Figure 57. Odd/Even Field Detection

18.5.6 INPUT PIXEL MEASUREMENT

The FLI106xx provides a number of pixel measurement functions intended to assist in configuring system parameters such as pixel clock, SDDS sample clocks per line and phase setting, centering the image, or adjusting the contrast and brightness.

18.5.7 IMAGE PHASE MEASUREMENT

The image phase measurement function measures the sampling phase quality over a selected active window region. This feature may be used when programming the source DDS to select the proper phase setting.

18.5.8 IMAGE BOUNDARY DETECTION

The FLI106xx performs measurements to determine the image boundary. This information is used to program the Active Window and center the image.

18.5.9 INSTANT AUTO

The FLI106xx supports Instant Auto for higher speed graphics good phase determination.

18.5.10 LETTERBOX DETECTION

Letterbox detection uses a firmware algorithm with hardware hooks to determine whether or not an input is letterbox format.

The algorithm should prove to be stable for high noise levels and in cases where a logo or text is present in the black bar regions of the image. The hardware hooks include top and bottom programmable windows.

18.5.11 VCR TRICK MODE

VCR design uses helical scan technology. The helical scan principal predominantly involves wrapping the tape around the drum in a helix creating diagonal tracks across the tape.

VCRs generally have four heads, two for SP speed and two for EP speed. The head pairs will each read an odd field and an even field and are spaced 180 from each other. The spacing between the EP head and the SP head is a few lines, such that the odd field EP head (EO) is near the even field SP head (SE) and vice versa.

In play mode, the tape rolls across the drum at speeds where one field is read with half a rotation of the head drum. As the tape rolls past the drum the heads read the material on the tape. The control track aligns the even head field to the even data field and the odd head field to the odd data field.

The VCR mode detector works on the principle that VCRs are unstable time bases. This means that there is some variation in the field size with respect to time. The number of asynchronous clocks per two adjacent fields (one frame) is counted and if this value changes, the input can be called a VCR. In Fast Forward or Rewind mode the number of lines per field changes by a considerable amount, therefore, counting lines per frame can yield a useful result.

19 VIDEO PROCESSING

19.1 NOISE REDUCTION AND SIGNAL CONDITIONING

19.1.1 3D / TNR NOISE REDUCTION

"3D" noise reduction is implemented by applying temporal (inter-field) and spatial (intra-field) noise reduction filters to the video data.

Temporal Noise Reduction (TNR) coefficients are a function of both Luma and Chroma channel motion values. In addition, FLI106xx has Look-Up Tables (LUTs) so that NR coefficients may be chosen based on the raw Luma value as well.

The temporal NR block for FLI106xx also supports Noise Reduction of the flesh tone regions. Due to Chroma motion detection in the NR block, color smearing can be avoided during NR.

A motion-adaptive frame-based recursive noise reduction is performed on both the chroma and luma data. An innovative noise meter measures the amount of noise in the backporch region without any serrations. Depending on the degree of noise present and motion sensitivity defined, the recursive filter values are selected. The degree of sensitivity to motion is programmable.

Facial features may be adversely affected if the noise reduction done on such areas uses the recursive filter levels optimized for the overall picture. To prevent this, and to provide a more natural picture, flesh tones are identified and when they are present, the noise reduction algorithm is modified. Furthermore, the TNR block helps to remove the noise without introducing image tearing artifacts.

19.1.2 CROSS-COLOR SUPPRESSION

The cross-color suppressor helps to eliminate objectionable cross-color artifacts produced when Composite NTSC or PAL signals are decoded. High frequency luma information leaks into the chroma channel during decoding to produce flashing colors or rainbow patterns.

The FLI106xx 3D comb filter removes these artifacts from composite inputs. For S-Video and Component video signals that do not use the 3D decoder, the Cross-Color suppression algorithm is the only way to eliminate this type of noise. The frame-based cross-color suppressor uses motion-weighted spatio-temporal filtering of the chroma to remove or significantly reduce the residual cross-color artifacts.

19.1.3 MEDIA NOISE REDUCTION

The media noise reduction block removes unwanted ringing and block noise from images that have undergone MPEG or JPEG compression and decompression. The two types of media noise that FLI106xx can reduce are:

- **Block Noise:** MPEG encoders, in the presence of an almost flat area, can create a squared structure due to the discrete 8x8 squares that are used in the MPEG compression process. This creates a noticeable squared structure in the image. The FLI106xx smoothens these square boundaries so they are not visible. The amount of smoothening is programmable between neighboring pixel values that will be smoothened. The amount of smoothening starts to decrease linearly from the maximum (defined by T0) to 0 (defined by 2 x T0) to avoid the hard switch effect of the smoothening applied.
- **Mosquito Noise:** The FLI106xx smoothens checker box and discrete noise artifacts referred to as "Mosquito Noise" around large edge transitions caused by MPEG encoders. The smoothening level is programmable.

19.1.4 REMOVAL OF CHROMA UP-SAMPLING ERROR

The FLI106xx has the ability to remove the Chroma Up-Sampling Error (CUE) or "Chroma Bug" found in many DVD players on the market today. This feature fixes the errors on many DVD players by up-sampling the 4:2:0 MPEG stream to a 4:2:2 sequence required for display video processing.

19.2 DCDI® BY FAROUDJA® VIDEO PROCESSING

19.2.1 FORMAT AND ASPECT RATIO CONVERSION—SCALING

The FLI106xx dual zoom/shrink scalers use an advanced scaling technique proprietary to Genesis Microchip Inc. They provide simultaneous high-quality scaling of real-time video images and graphics on both channels. An input field/frame is scalable arbitrarily in both the vertical and horizontal dimensions.

19.2.1.1 ASPECT RATIO CONVERSION AND NON-LINEAR SCALING

The input image is separated into three zones horizontally: left, center, and right. The center zone is scaled at a programmable fixed ratio. The left and right zones have a programmable changing scale factor that changes from left to right (see Figure 58 below). The scale ratio change can either be linear or parabolic.

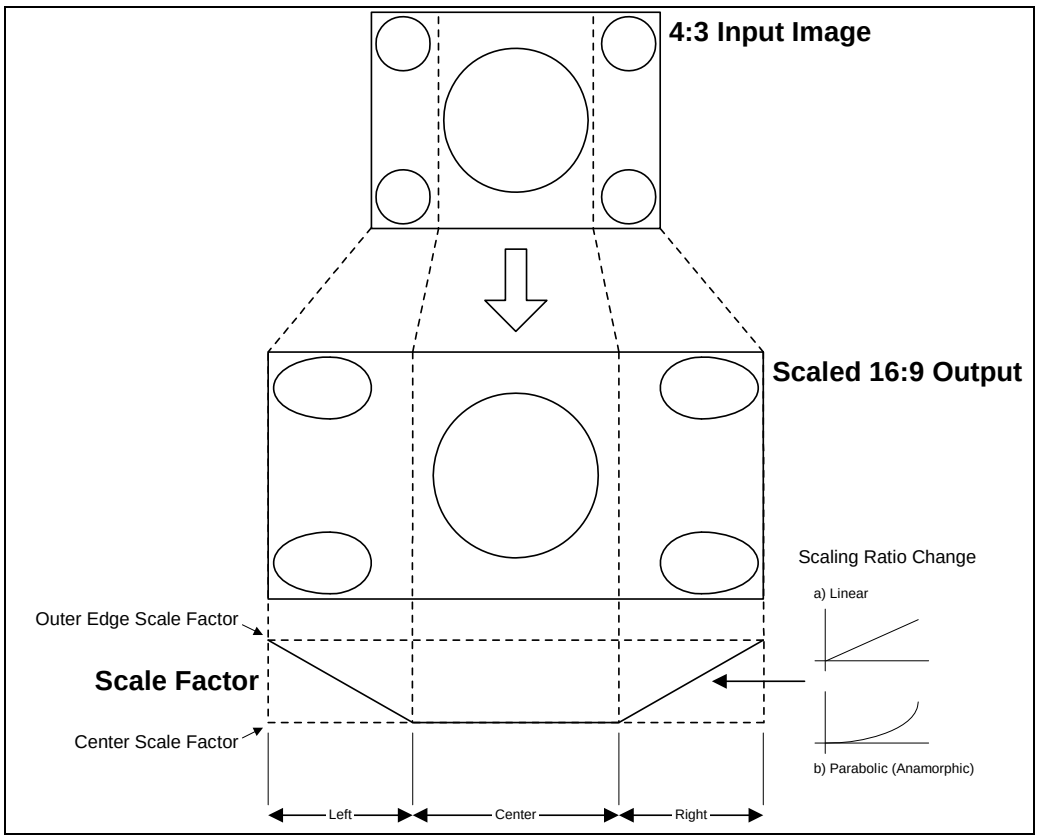


Figure 58. Non-Linear Scaling of a 4:3 to 16:9 Aspect Ratio Conversion

A conversion for 16:9 to 4:3 aspect ratio would use the same method with an inverse parabolic ratio for the scale factor of the left and right zones.

19.2.2 DYNAMIC SCALING AND ANIMATION

The FLI106xx has an advanced scaling engine that can support dynamic scaling of video without producing any judder or dropped frames during the video sequence. Advanced synchronization hardware aligns the video to produce smooth resizing of the image.

With split screen dynamic scaling, two windows are shown on the display (as with Picture by Picture [PBP]) with the ability to change the size of the selected window through +/- keys on the remote control.

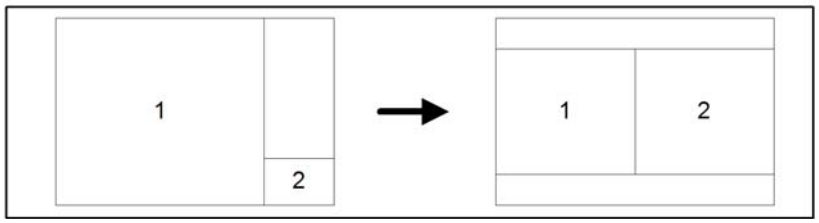


Figure 59. Example of Split Screen Dynamic Scaling

19.2.3 INVERSE 3:2 / 2:2 PULL-DOWN DE-INTERLACING

Video graphics with 3:2/2:2 field sequences are processed with the built-in Adaptive Film Mode controller. The field sequence is first detected by examining the field motion values and once a field sequence is identified, the corresponding film mode processing is applied.

19.2.4 ADAPTIVE MEDIA DISPLAY (3:3 AND 2:2 FILM MODES)

The FLI106xx has the ability to take 3:2 film sequences of 24 Hz original film content and convert this to an output of 48 Hz or 72 Hz vertical refresh rate. This feature reduces the noticeable uneven judder seen on a 3:2 film sequence that is converted to 60 Hz vertical refresh rate.

19.2.5 MOTION ADAPTIVE DE-INTERLACING (MADI)

Motion adaptive de-interlacing in the FLI106xx is a pixel-based two-phase process. Phase 1 involves the detection of motion and the generation of a motion value for each pixel. These pixel motion values are used as a measure of the current "degree of motion". In Phase 2, the pixel motion values are used to select the appropriate de-interlacing technique.

As a result, areas of an image that are not moving will be fully static (flicker free) and moving objects will have smooth edges.

19.2.6 DCDI® BY FAROUDJA LOW-ANGLE DIAGONAL INTERPOLATION

In addition to the advanced de-interlacing capabilities mentioned in the previous sections, further image enhancement is achieved by applying special processing to a moving low-angle diagonal pattern in a video image.

For motion video, the intra-field interpolation is done using the patented and highly acclaimed DCDi® (Directional Correlational De-interlacing) algorithm by Faroudja®. Conventional interpolation algorithms operate on the basis that the current pixel is related to the pixels above and below it. This is untrue of diagonal edges. Whenever an edge is not vertical, the way the current pixel depends on the angle of the edge is related to those diagonally above and below it. Hence, conventional vertical interpolation algorithms work well on edges close to the horizontal and vertical directions. However, they will break down completely as the angles of edges become more diagonal, causing jagged edge artifacts. DCDi® by Faroudja computes and tracks the angles of edges and uses this information to optimally fill in the missing pixels, removing jagged edge artifacts.

19.3 ADVANCED DIGITAL COLOR CONTROLS

The advanced digital color controls consist of the block in the following diagram:

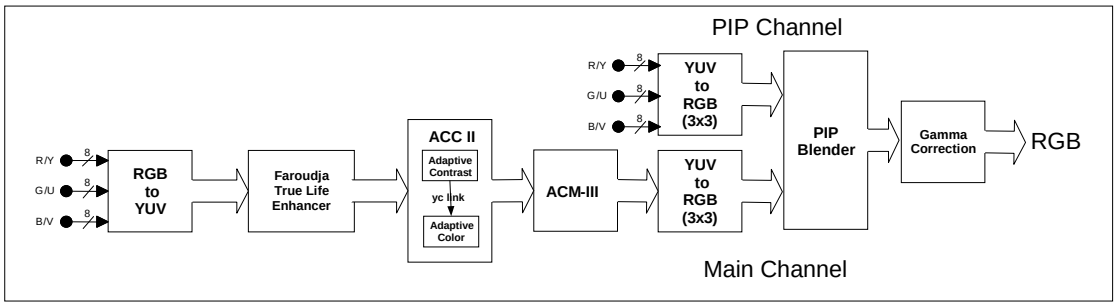


Figure 60. Advanced Digital Color Control Block

19.3.1 ADAPTIVE CONTRAST CONTROL II (ACC II)

Most video content is tailored for display on CRTs or in movie theatres. However, CRT monitors have a wider dynamic range than LCD monitors. Therefore, it is desirable to enhance the dynamic range when video is displayed using LCD monitors.

ACC II enhances the contrast of the image. This feature makes dark colors darker and bright colors brighter. In addition, the contrast enhancement is adaptive. That is, dark scenes are transformed to preserve color resolution of darker colors and bright scenes are transformed to preserve the color resolution of brighter colors. This is done by calculating a running average of the luminance content of the image, divided up into three ranges. The histogram is then used to select the weighting coefficients between three different luminance transfer functions. The running average may be applied over a programmable number of frames.

ACC II can be applied within a highlight window or over the full display area.

19.3.2 ACTIVE COLOR MANAGEMENT–3D™ (ACM-3D)

ACM-3D provides TV style control of global color parameters like hue, saturation, and contrast, and local color changes such as skin tone adjustment or green enhance.

ACM-3D can be applied within a highlight window or over the full display area. The ACM-3D feature for FLI106xx provides three dimensional color controls for Y, U, and V signals for a particular color component.

19.3.2.1 FLESH TONE CORRECTION

A function of the ACM-3D block, the dedicated Flesh Tone correction feature has been added to ensure flesh tones are reproduced accurately. Without flesh tone compensation, some video processing techniques may manipulate flesh tone shades incorrectly.

19.3.2.2 BLUE STRETCH

Another function of the ACM-3D Block is the Blue Stretch system. The Blue Stretch system analyzes and manipulates the white point of the incoming signal in order to shift the color temperature towards blue. A bluer white point is considered beneficial by some television viewers. The amount of Blue Stretch is programmable.

19.3.2.3 SUPPORT FOR OVERLAPPING REGIONS

This feature allows the 6-axis color controls to be controlled individually for Hue, Saturation, Offset, and Gain. To provide higher contrast resolution, the FLI106xx supports both 3-bin (automatic) and 5-bin (automatic) modes.

19.3.3 COLOR CONTROLS

The FLI106xx provides high-quality digital color controls that can be applied independently to the video and RGB data streams. These consist of a full 3x3 RGB matrix multiplication stage, followed by a signed offset stage as shown below in Figure 61.

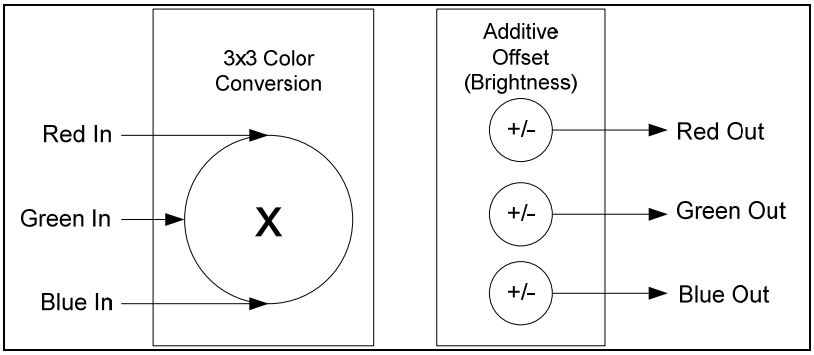


Figure 61. RealColor™ Digital Color Controls

This structure can accommodate all RGB color controls such as contrast (multiplicative stage) and brightness (signed additive offset). In addition, it supports all YUV color controls including brightness (additive factor applied to Y), contrast (multiplicative factor applied to Y), hue (rotation of U and V through an angle), and saturation (multiplicative factor applied to both Y and V).

To provide the highest color purity, all mathematical functions use 12 bits of accuracy to ensure that 10 bits of video data is maintained. The final result is dithered to 10, 8, or 6 bits (as required by the LCD panel).

19.3.4 GAMMA LOOK-UP TABLE (LUT)

The FLI106xx provides a 10- to 12-bit Look-Up Table (LUT) for each input color channel intended for Gamma correction and to compensate for a non-linear response of the LCD panel. A 12-bit output results in an improved color depth control. The 12-bit output can be used by TTL (YUV 4:2:2) or by the LVDS output interface by dithering to 10, 8, or 6 bits (see Pane below). The LUT is user-programmable to provide an arbitrary transfer function. Gamma correction occurs after the zoom/shrink scaling block.

19.4 MAIN CHANNEL VIDEO ENHANCEMENT

19.4.1 MAIN CHANNEL SHARPENING FILTERS

The FLI106xx provides a sharpness system for both the Luminance and the Chrominance based on filtering. It includes a programmable noise coring function which is used to control noise amplification as well as a non-linear shaper to control the sharpness level in different amplitude ranges. The amount of sharpness is controlled by the independent programmable horizontal and vertical gains.

The sharpness system for FLI106xx supports advanced peaking filters which help to improve sharpness without increasing noise.

19.4.2 PIP CHANNEL SHARPENING FILTERS

The sharpening filters are duplicated on the PIP channel which allows the user to independently enhance the second channel video to a level different from the Main channel. The PIP channel also contains a programmable noise coring function.

19.4.3 FAROUDJA® TRUELIFE™ NON-LINEAR ENHANCEMENT

The Faroudja® TrueLife™ Non-Linear Enhancer provides a non-linear edge enhancement of both Chrominance and Luminance to give video images a more realistic and life-like appearance.

The system provides independent controls for "small" and "large" horizontal and vertical Luminance edge enhancement as well as for a horizontal Chrominance edge enhancement. The system includes a noise coring function that is controlled by means of a "Noise Threshold".

The edge enhancement engine can be combined with the Sharpness Filters engine to increase flexibility in the customization of the sharpness system.

19.5 PICTURE-IN-PICTURE BLENDER—PIP CHANNEL PROCESSING

The FLI106xx allows for very flexible PIP display configurations whereby either the graphics or the video channel may act as the PIP source to overlay over the other channel. Any one of the inputs (analog RGB, 30/24-bit digital, 10/8-bit digital, YPbPr, composite video, etc) may be multiplexed to either channel. The PIP can be from any of the Analog Input Ports (AIP 1 or 2) or Digital Input Port (DIP).

Single PIP allows the PIP display to be placed arbitrarily in the display window. It can be placed within the Main display, partially overlapped with Main display, or fully detached from Main display. The size of the PIP display is fully programmable. Also, single PIP display allows 16 levels of alpha blending within the PIP window either with a specified background color or the Main channel. A special case of single PIP display is side-by-side configuration.

Using external memory, the second channel image data can be frame-rate converted so that it matches the refresh rate of the main image and can subsequently be merged in various picture arrangements. The frame rate conversion of the second channel ensures no frame tear is present in the PIP channel video.

The PIP channel contains advanced video processing including DCDi low angle processing and sharpening filters to provide a high quality image, even on the second channel.

Note: Only one of the input channels (Main or PIP) can support inputs through the video decoder for the FLI106xx. Either of the channels can act as Main or PIP based on the swap feature.

19.5.1 PIP MATRIX SUPPORT

Two PIP matrices for FLI106xx are revealed in the tables below:

Table 39. FLI106xx PIP Matrix

		Main-Channel						
		HDMI	**CVBS**	**Component YPbPr**	**PC RGB**	**Tuner Digital**	**Tuner Analog**	**VXI**
Sub-Channel	**HDMI**	N/A	Yes	Yes	Yes	Yes	Yes	Yes
	CVBS	Yes	N/A	No	No	Yes	No	Yes
	Component YPbPr	Yes	No	N/A	No	Yes	No	Yes
	PC RGB	Yes	No	No	N/A	Yes	No	Yes
	Tuner Digital	Yes	Yes	Yes	Yes	N/A	Yes*	Yes
	Tuner Analog	Yes	No	No	No	Yes*	N/A	Yes
	VXI	Yes	Yes	Yes	Yes	Yes	Yes	N/A

*Note: PIP for Digital Tuner and Analog Tuner inputs is only possible with dual-tuner can configuration.

19.5.2 PIP BORDER AND BLENDING

The PIP display mode consists of two windows with one smaller window sitting on top of the other. This smaller window is called the PIP window. Its size and position are fully programmable up to the entire display size. The transparency (blend) level of the PIP window is adjustable up to 16 levels.

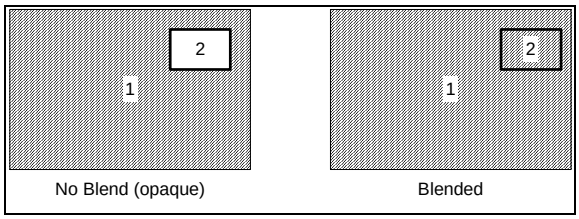


Figure 62. Example of PIP

The FLI106xx has hardware support for the PIP border. In single and multi-PIP modes, the PIP border size, color, and on/off status is programmable for each PIP window. An independent highlight window can also be programmed to be used primarily for highlighting the active PIP window in a multi-PIP system.

19.5.3 COLOR KEY OVERLAY

In Color Key Overlay mode, data from one channel (overlay) is overlaid on top of another using color keying. Colors below a programmable threshold in the overlay channel are considered transparent. This mode enables an external OSD to be overlaid on top of another data stream (e.g. a video image).

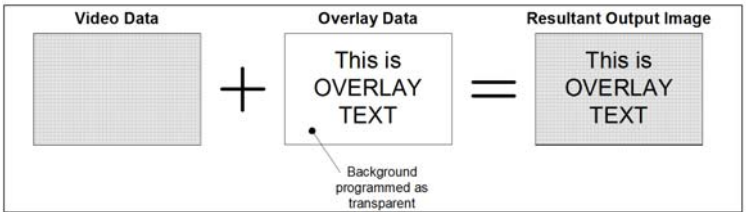


Figure 63. Color Key Overlaying Example

The overlay data can come from either the Main or secondary channels with the other channel providing the Picture or Image data.

20 DMA AND TRANSACTION CONTROL

The FLI106xx DMA supports 32 main channels and 64 link channels. Link channels are specified in the main channel and link channel transfers are initiated after the completion of main channel transfers. Each DMA channel provides its set of source, destination, transfer size, and address control registers and can be independently configured to perform a DMA transaction. Channels 0-17 can also be used for associated peripheral devices. These channels support handshake with the on-chip peripheral block allowing the corresponding peripheral devices to initiate DMA transfers in chunks that are suited for the peripheral block. Even channels 0-17 can be used according to software needs.

The DMA controller consists of the following components:

- Channel Configuration Table
- Channel Link Table
- Event Handler
- Configuration Bus Interface
- Scheduler
- Address Generation logic

The following figure shows the internals of the DMA controller.

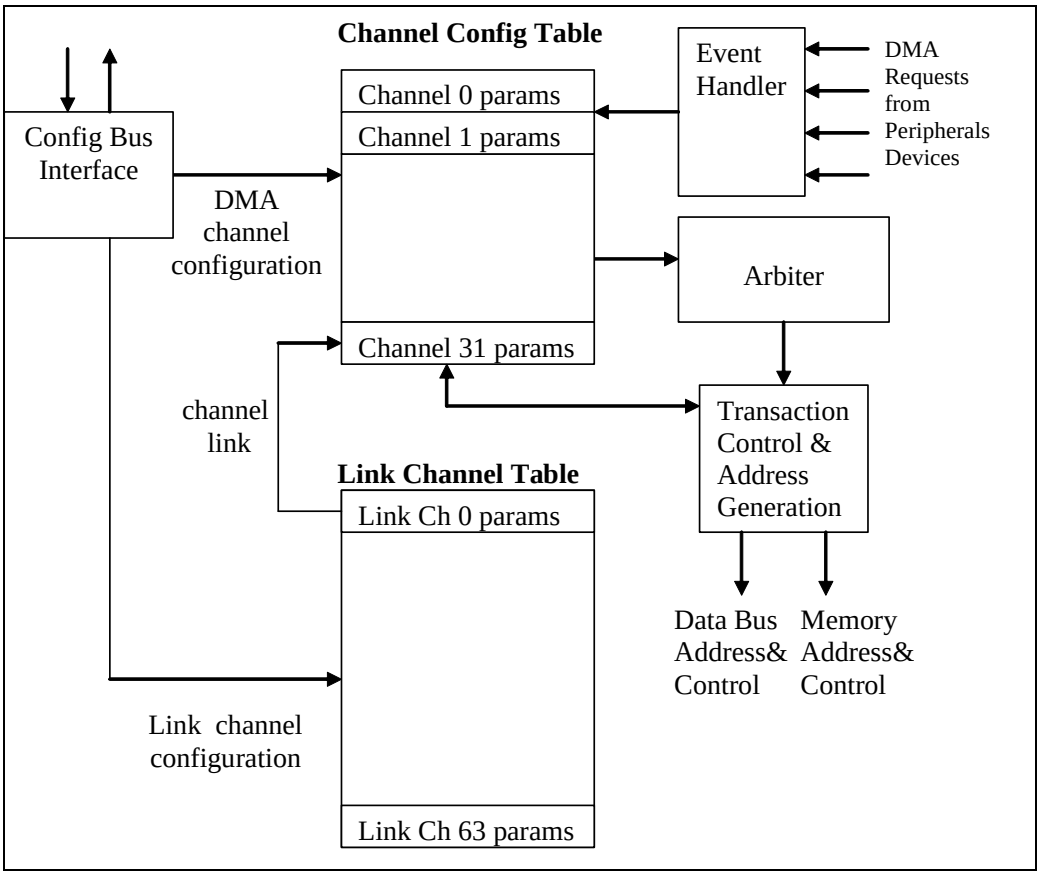


Figure 64. DMA Controller Architecture

All four processors can program DMA channels via memory mapped configuration registers that are local to each processor. The 2D Engine can also program its allocated DMA channels based on its own programmed commands. To configure a channel, MIPS/MPEx/DE2D writes channel parameters to any one or all five of the DMA configuration registers. Each processor specifies

the DMA channel or link number and indicates whether the operation is read or write. Channel parameters (defined in Section 20.1) include:

- Source and destination starting address
- Number of elements per set
- Element offset in a set
- Number of sets in a transfer
- Set offset

The content of these registers is then transferred by hardware to the corresponding entry of the channel configuration/channel link table for a write, or copied from the channel configuration/channel link table to DMA configuration registers for a read.

All DMA transactions are always performed in multiples of 32-bit words at word-aligned addresses. DMA supports long transfers without software intervention. DMA can subdivide programmed channel transfers into specified block size and can switch channels at block boundaries.

DMA supports linear and 2-dimensional transfer with programmable offsets between elements and sets. Each DMA channel is capable of performing transfers from a memory (DRAM) or any Data Bus client source to either a memory or a Data Bus client destination.

20.1 CHANNEL TRANSFER PARAMETERS

Figure 65 summarizes the terminology used to program a DMA channel and gives an example of 1-dimensional and 2-dimensional transfers. The full transfer requested by 1 channel configured one time by software is a **DMA Transaction**.

- **Element** is the basic unit of a DMA transfer and is a 32-bit word. All DMA transactions are always performed in granularity of 32-bit.
- A **Set** is a linear group of continuous or interleaved elements. A set may correspond to a single line of elements of a 2-D structure, or a subset of a 1-D structure with a fixed offset between elements.
- Set has **ELEMENT_COUNT** number of elements. ELEMENT_COUNT specifies the number of elements in a set. ELEMENT_COUNT can be between 1 and 4095.
- **Element_Offset** defines the delta between two consecutive elements.
- A transfer consists of a **SET_COUNT** number of sets. SET COUNT specifies the number of sets comprising a single DMA transfer. SET COUNT can be between 1 and 4095.
- **Set_Offset** defines a skip distance between a first element in one set and the first element in the next set in 32-bit words.

DMA does not schedule a DMA transaction as a single atomic transfer. The size of the atomic transfer is specified as the BLOCK SIZE for that channel. BLOCK_SIZE is independent of Set, is not aligned to a Set, and may span one or more Sets. The last Block transfer of a DMA transaction is shorter than Block Size if the total number of elements is not divisible by BLOCK_SIZE.

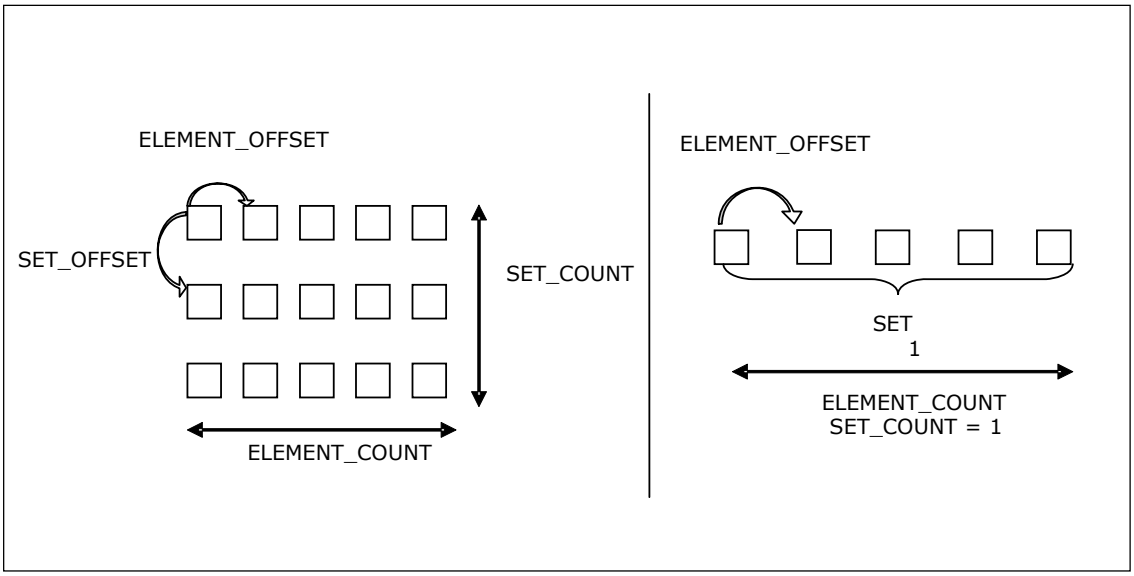


Figure 65. DMA Configuration Terminology

20.2 CHANNEL CONFIGURATION TABLE

The Channel Configuration Table contains the following parameters for each main channel:

- SOURCE_ADDRESS
- DESTINATION_ADDRESS
- SOURCE_ELEMENT_OFFSET
- DESTINATION_ELEMENT_OFFSET
- SOURCE_SET_OFFSET
- DESTINATION_SET_OFFSET
- ELEMENT_COUNT
- SET_COUNT
- OPTIONS

Reading a channel's parameters while a channel is being serviced does not return the programmed values of parameters. Each DMA channel can be simultaneously configured and serviced. A DMA channel can point to any of the 64 link channels. If it does, that channel is reloaded from the pointed link channel upon completion. When a channel is reloaded, the entire contents of a link channel are copied to the current DMA channel configuration table and may immediately request a DMA transfer.

20.3 CHANNEL LINK TABLE

The Channel Link Table contains 64 entries. Link table entries are identical to channel configuration table entries. Unlike Channel Configuration Table entries, links are merely configuration information storage. They cannot request DMA service. The link table allows software to create a number of ordered (linked) DMA transfers at once and not be interrupted after each stage.

20.4 CHANNEL SCHEDULER

The DMA Channel Scheduler implements four levels of priority, with round-robin scheduling within one priority level. In addition, after four requests have been scheduled at a particular priority level, one request is scheduled from lower priority levels. For example, after four requests scheduled at priority 3 the scheduler schedules one request at priority 2, and after four

schedules at priority 2 (and 16 at priority 3), it schedules a priority 1 request. This scheme allows some progress for lower-priority channels in the presence of higher-priority ones. There is no wastage of unused slots: transactions may be scheduled back-to-back.

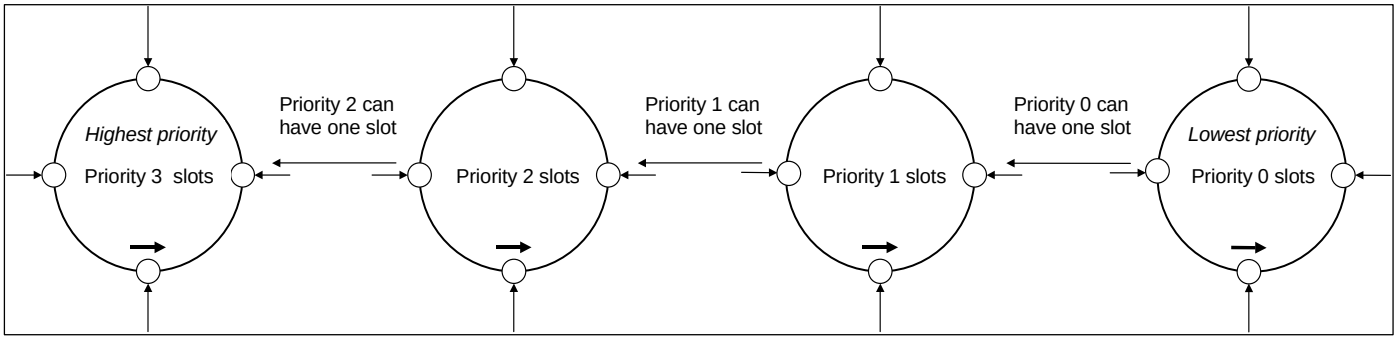


Figure 66. Transaction Scheduling

20.5 HARDWARE TRIGGER EVENTS

The FLI106xx DMA supports DMA Transfer Requests by various peripheral devices (termed "DMA Trigger Events"). Each DMA Request results in transfer of 1 Block (configured between 4 and 512 32-bit words). Multiple outstanding DMA Requests may be issued.

The on-chip peripheral device issues a DMA Request when the DMA needs to transfer data to or from the peripheral. When the DMA Request is received by an Event Handler, the corresponding DMA channel is activated and waits for its turn to be scheduled for DMA transfer. Channels 0-17 have associated peripheral requests. However, these channels can be used for non-peripheral (i.e. software-driven) transfers, as well.

20.6 DMA CONFIGURATION BUS

DMA channel and link tables are accessed by the processors and 2D Engine over a shared DMA Configuration Bus, which is arbitrated for automatically in hardware. The software running on the processors programs its local set of DMA configuration registers. These parameters are transferred over the DMA Configuration Bus to the specified entry of the DMA channel/link table. If the operation is a read, hardware copies the corresponding configuration or link table entry into MIPS-mapped DMA configuration registers.

20.7 TRANSACTION CONTROL AND ADDRESS GENERATION

This block generates source and destination addresses and control information for a channel selected by the DMA Scheduler.

20.8 DMA CHANNEL ALLOCATION

The FLI106xx DMA supports 32 Main and 64 link channels. Although allocation of the channels to CPUs is not enforced in hardware, there are certain recommended allocations based on the application being targeted. However, association of peripheral blocks to channels is fixed through mapping of their hardware trigger events to specific channels.

The recommended DMA channel allocation for single HD Decode is shown in the following table. Channels 0-12 and 16-17 are associated with particular peripheral blocks and receive hardware DMA requests (events) from these blocks. These channels can be used for other functions if the associated blocks are disabled. Channel 31 is reserved for debug. All other channels are general-purpose and can be reassigned according to system needs.

Table 40. Single HD Decode DMA Channel Allocation

Channel	Function	Number of Links	Fixed Allocation
0	Audio Output DMA Buf0 (AUDO)	2	Y
1	Audio Output DMA Buf1 (AUDO)	2	Y
2	TS input from CDI FIFO (TDMX)	2	Y
3	MPEG VLD DMA (MVP)	0	Y
4	Video Out Luma (MDTP)	0-3	Y
5	Video Out Chroma (MDTP)	0-3	Y
6	GFX Out (VODP_GDP1)	0-2	Y
7	DES Output FIFO (TDMX)	0-2	Y
8	2D source read (DE2D)	3, #55, 58, 61	Y
9	2D destination read (DE2D)	3, #56, 59, 62	Y
10	2D destination write (DE2D)	3, #57, 60, 63	Y
11	DES Input FIFO (TDMX)	0	Y
12	Host Bus (CCHB)	0	Y
13	CDO (Douglas)	2	Y
14	Audio In Ch0 (AUDI)	2	Y
15	Audio In Ch1 (AUDI)	2	Y
16	Audio In Ch2 (AUDI)	2	Y
17	Reserved	2	Y
18	GDP2 (Douglas)		Y
19	GDP3 (Douglas)	2	Y
20	SW Allocation [e.g., MPE2 (Predictor fetch setup: FWD Luma & Chroma / top/bottom field)]	3	N
21	SW Allocation [e.g., MPE2 (Predictor fetch setup: BWD Luma & Chroma / top/bottom field)]	3	N
22	SW Allocation [e.g., MPE2 (Predictor fetch running: FWD Luma & Chroma / top/bottom field)]	3	N
23	SW Allocation [e.g., MPE2 (Predictor fetch running: BWD Luma & Chroma / top/bottom field)]	3	N
24	SW Allocation [e.g., MPE2 (Writeback setup: Luma & Chroma / top/bottom field)]	3	N
25	SW Allocation [e.g., MPE2 (Writeback running: Luma & Chroma / top/bottom field)]	3	N
26	SW Allocation [e.g., MPE0 (Code and data)]	SW-dependent	N
27	SW Allocation [e.g., MPE1 (code and data)]	SW-dependent	N
28	SW Allocation [e.g., MPE1 (code and data- backup for high transfer rates)]	SW-dependent	N
29	Available to SW	SW-dependent	N
30	Available to SW		
31	Debug, Available to SW		

21 MEMORY CONTROLLER

The FLI106xx memory controller transfers data between the off-chip Double Data Rate (DDR) DRAM and the on-chip Client Queues (FIFOs) that require memory access. The memory controller services its clients: MIPS CPU, Data Bus DMA Controller, Video subsystem clients, and AHB Bus Bridge. The bus width of the DDR DRAM is 16/32 bits.

The Memory Controller supports DDR2 DRAM compliant to JESD79-2 for DDR2-800 (for 400MHz).

Note: When FLI106xx is used in 16-bit mode, it must use 64MB of the 400MHz DDR2. 16-bit DDR2 mode can only support WXGA panel

The table below identifies the specific configurations supported.

Table 41. DRAM Devices Supported by FLI106xx

Manufacturer	Device Size / Max Speed	Configurations
Micron, Samsung, Hynix, Infineon, Elpida, Toshiba (and others that are compliant to JEDEC std JESD79C)	256Mb / 200-400MHz-DDR2	1 x (16M x 16)
	256Mb / 200-360MHz-DDR2	2 x (16M x 16)
	512Mb / 200-400MHz-DDR2	1 x (32M x 16)
	512Mb / 200-360MHz-DDR2	2 x (32M x 16)

Note: Manufacturers/Part numbers not listed here may be supported. Please check with your Genesis representative.

Features:

- Supports industry standard Double Data Rate (DDR)
- DRAM from 128Mbit to 1Gbit device sizes (512 MBit for DDR2)

  Note: FLI106xx cannot support 1Gbit DDR2 device
- High-performance memory controller with application specific optimizations for bandwidth efficiency
- Programmable DRAM data width and user word size
- Programmable DRAM access timing parameters and burst length
- Automatic refresh generation with programmable refresh intervals
- Self-refresh mode to reduce system power consumption
- Supports CAS latency of 2/3/4/5/6 clk
- Supports four priority levels (0 - lowest, 3 - highest)
- Support Byte Mask for writes
- Support MIPS bypass of memory controller queues for shortest latency
- Support 16-bit and 32-bit DDR interface (Only supports x16 DDR/DDR2 memory)

This DDR Controller allows the user logic to simply read from or write to the memory system without having to be concerned with specific DRAM control and timing issues.

The DRAM timing, such as row and column latency, pre-charge timing, and row access length, are programmed to their default values at system reset for a DDR400 device. They can also be reprogrammed before initialization if the user wishes to change them to optimize system performance.

21.1 MEMORY CONTROLLER CLIENTS

The Clients' Interface to MEMC is synchronous to MEM_CLK (memory controller clock). Clock domain crossing to the clients' clock domain is handled by each client. The following clients are handled by this block

1. Data Bus (Dbus) Read
2. Dbus Write
3. Video Subsystem Memory Arbiter Read
4. Video Subsystem Memory Arbiter Write
5. MIPS Read
6. MIPS Write
7. AHB Bus Read/Write

21.2 MEMORY MAP

All FLI106xx agents view a single Address Map. In general, address bit A[28] selects between DRAM and Data Bus access, and address bits A[27:20] must be decoded to select one device on Data Bus.

The address maps for devices are listed below:

Table 42. System Address Map

Device	Address Range	Size
DRAM	0000_0000h – 0FFF_FFFFh	256MB
AUDP (MPE0)	1000_0000h – 10FF_FFFFh	16 MB
TDMUX (MPE1, CDI FIFO)	1100_0000h – 11FF_FFFFh	16 MB
MVP (MPE2 & MPEG Decoder)	1200_0000h – 12FF_FFFFh	16 MB
Reserved	1300_0000h – 130F_FFFFh	1MB
Reserved	1310_0000h – 131F_FFFFh	1MB
Reserved	1320_0000h – 132F_FFFFh	1MB
Reserved	1330_0000h – 133F_FFFFh	1 MB
Reserved	1340_0000h – 137F_FFFFh	4 MB
AUDO Audio Output	1380_0000h – 138F_FFFFh	1 MB
AUDI Audio Input	1390_0000h – 139F_FFFFh	1 MB
Pbus	13A0_0000h – 13AF_FFFFh	1 MB
Reserved	13B0_0000h – 13BF_FFFFh	1 MB
Reserved	13C0_0000h – 13FF_FFFFh	4 MB
MIPS EBB registers	1400_0000h – 147F_FFFFh	8 MB
DE2D	1480_0000h – 14FF_FFFFh	8 MB
MDTP DMA data fifo, VODP_GDP DMA data fifo (Mpeg display timing processor, VODP graphics display processor)	1500_0000h – 157F_FFFFh	8 MB
VBI SRAM	1580_0000h – 158F_FFFFh	1 MB
CCHB FIFO	1590_0000h – 159F_FFFFh	1MB
CDO Data FIFO	15A0_0000h – 15AF_FFFFh	1MB
Reserved	15B0_0000h – 15FF_FFFFh	5 MB
Pattern Generator	1600_0000h – 167F_FFFFh	8 MB
Reserved	1680_0000h – 17FF_FFFFh	24 MB
AHB	1800_0000h – 1BFF_FFFFh	64 MB
Host IF bus	1C00_0000h – 1FFF_FFFFh	64 MB

The Memory Controller arbitrates requesters on block (burst of requests) granularity.

Memory requests to different banks may be serviced out of order. The Memory Controller supports data reordering so that data is returned to the requestor in the order it was requested.

High priority MIPS CPU requests (cache misses and un-cacheable accesses) bypass memory controller queues for the shortest possible latency. This significantly improves software performance on the MIPS CPU.

The FLI106xx Memory Controller supports a Self Refresh DRAM mode, which is useful for standby/power down implementation.

21.3 DRAM DATA ORGANIZATION

The following waveform explains how a double word is written to SDRAM when two devices of x16 width are connected. Note that these diagrams represent only the functionality and not the true timing waveforms for DDR write operation.

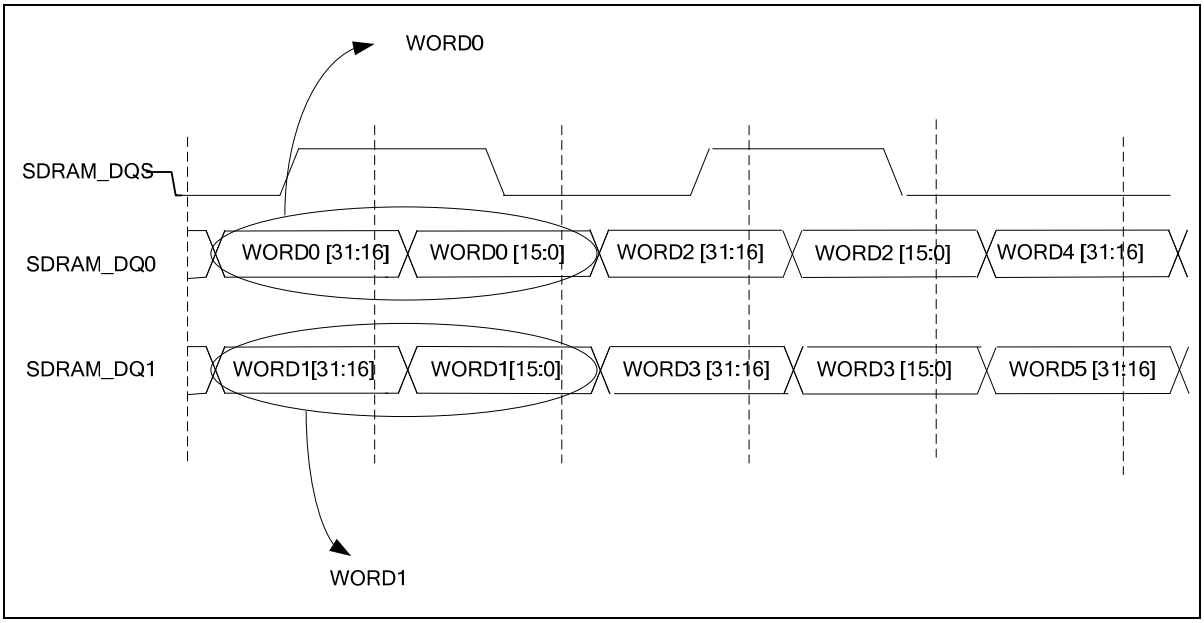


Figure 67. Data Word Transfer on x16x2 SDRAM Data Bus

The following waveform shows how a double word is written to SDRAM when a single device of x16 width is connected.

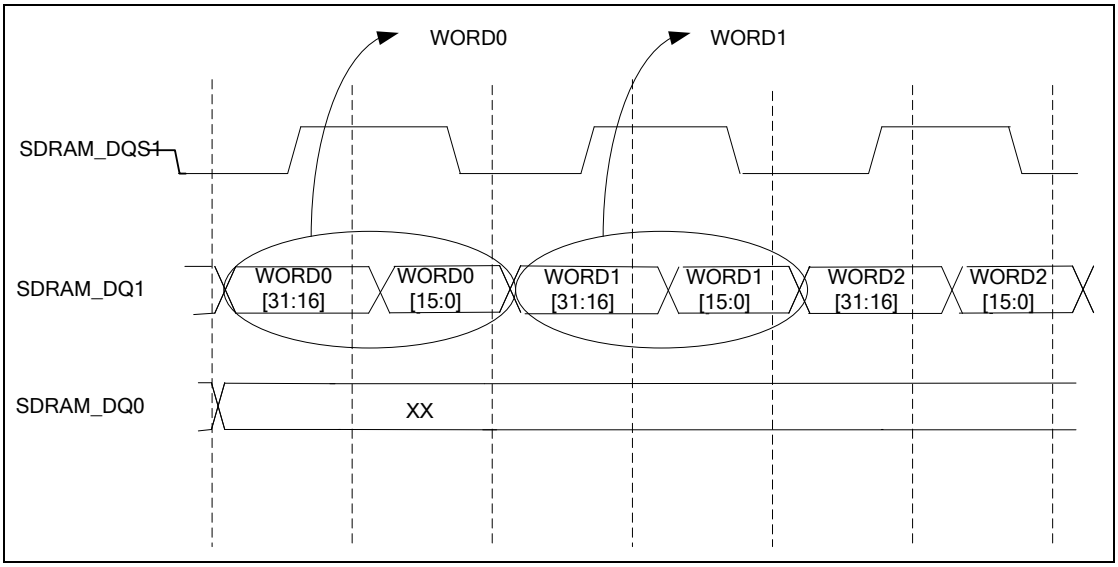


Figure 68. Data Word Transfer on x16x1 SDRAM Data Bus

22 GRAPHICS ACCELERATION

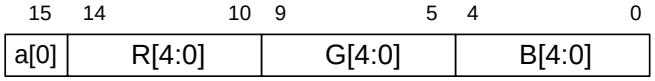
The FLI106xx supports high performance 2D graphics for rendering closed captioning and advanced Graphical User Interfaces (GUIs) for EPGs and setup menus. The 2D Engine (DE2D) can perform graphics operations on rectangular areas of a graphics frame stored in memory. It supports 1/2/4/8/16 and 32 bits per pixel formats. DE2D can be further used for rendering and composing closed captions (as per EIA 608/708) and subtitles (as per ETSI EN 300 743 V1.2.1 Standards).

The DE2D data is Source and Destination. The Source data is always read from memory and written to the Source FIFO. The Destination data is R/W with Dest Read FIFO serviced by the second DMA and a DestWrite FIFO serviced by the third DMA channel (using the byte mask). The Destination data read and write are always to/from the same block of addresses. The Destination data format may be either an 8-bit CLUT index or 16-bit aRGB (1:5:5:5) format or 32-bit aRGB (8:8:8:8). The 8-bit index format is expanded to 32bpp using CLUT in the Graphics Display Processor.

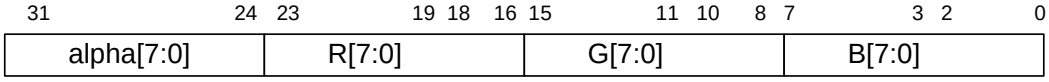
Table 43. Destination Formats

Destination Format	Description
8	8-bit index (to Graphics Display Processor CLUT)
16	aRGB (1:5:5:5)
32	aRGB (8:8:8:8)

The 16bpp pixel layout is as follows:



The 32bpp pixel layout is as follows:



Note: The OSD channel of the video processor block must support the aRGB formats and use the "alpha" bit to enable OSD pixel transparency when combining OSD with Video.

The Destination need not always be the actual display area. The DE2D could be used to construct small images in non-display memory (with a simple linear addressing) to be used as cached objects that are later rasterized and copied to the display. For example, a set of text characters may be created in non-display memory so that they may be used repeatedly for on-screen text writes.

The Source data may be either a compact color format of 1, 2, 4 or 8bpp, or in the same format as the Destination data. When the Source data is defined as a compact format, it will be "color-expanded" to the display color depth before being combined with the Destination data. In addition, YCbCr 4:2:2 format is supported for graphics blend with JPEG. 8-bit alpha with fixed R, G, and B components is supported to speed up anti-aliased font rendering.

Note: FLI106xx DE2D supports 8-bit index Color Expand with 256-entry CLUT.

Table 44. Source Formats

Source Format	Description
1-bit	1-bit color expand (via DE2D CLUT)
2-bit	2-bit color expand (via DE2D CLUT)
4-bit	4-bit color expand (via DE2D CLUT)
8-bit	8-bit color expand (via DE2D CLUT)
16-bit	aRGB (1:5:5:5)
32-bit	aRGB (8:8:8:8)
8-bit alpha	Expanded to aRGB 8:8:8:8 or 1:5:5:5
YCbCr16	4:2:2 pixel in Cb Y0 Cr Y1 format, interpolated to 4:4:4 and converted to aRGB (8:8:8:8)

All DE2D operations (Drawing Commands) will be programmed as a set of DE2D controls plus the variable configuration data for the three DMA channels. A dedicated Command Interpreter block will translate this Command to the DMA configuration register format.

The destination for both Read DMA channels are the Source and Destination FIFOs; the source for the Write DMA channel is always the Dest FIFO. All DMA element offsets will always equate to 1. The data FIFOs will not have any Set Offset. The Start Address, Counts, and Offsets for Dest Write DMA are the same for Dest Read DMA.

The DE2D does any and all required pixel and bit-map alignments. The DMA channels only move data to/from memory on 32-bit aligned addresses and always in elements of 32-bits.

The DE2D pixel alignment operations do shifting of data at the Source FIFO only. Destination reads and writes do not require any shifting. Destination writes require pixel write-masking to handle the alignment (clipping) of the Destination data at the pixel level within each 32-bit Element.

Although most drawing operations logically work in pixel X/Y coordinate representation, the DMAs will still be programmed in actual Start address, Set count, Element count, Set Offset.

The DE2D is serviced by three dedicated DMA channels. The DMA channels are responsible for supplying data to and taking data from the DE2D. The DE2D performs operations on just the Source area or both Source and Destination area, and writes back the result of the operation to the Destination area. The DMA channels are similar to other DMA channels with one small difference; the DMA channel that gets DE2D data to write back to memory also gets a 4-bit byte write-mask for each 32-bit word received. The other two DMA channels are only used to read data from memory and give the data to the DE2D. All DE2D drawing commands can be programmed as a set of 2D engine controls plus the Source and Destination rectangle configuration. A dedicated Command Interpreter block translates this command to the DMA configuration register format. The Source and Destination for the Read DMA channels are the memory and the Internal Read FIFO, respectively. For the Write DMA channels, the Source is the Destination Write FIFO and the Destination is memory.

The DE2D functions are driven by the following two components:

- **Command Interpreter:** Executes command list processing
- **Data Engine:** Processes the drawing requested using the commands

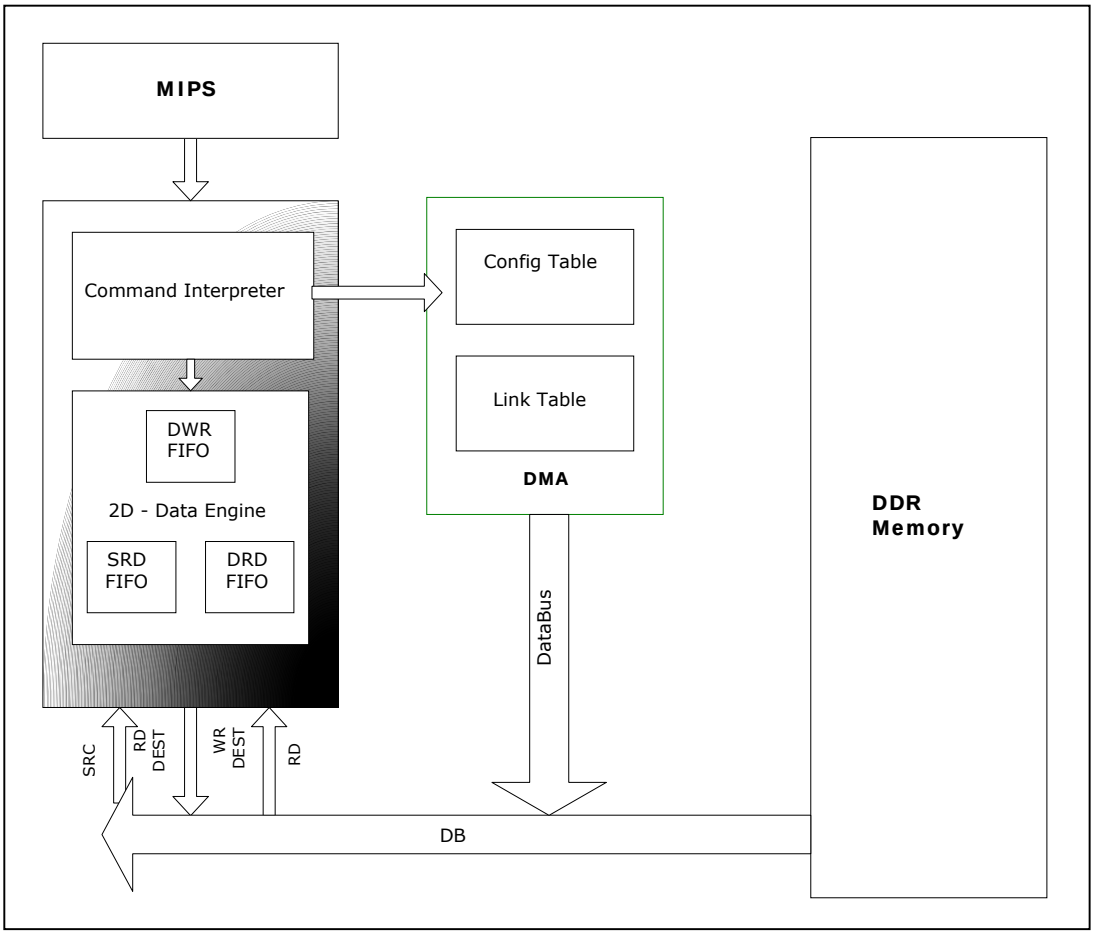


Figure 69. DE2D Interface Diagram

22.1 COMMAND INTERPRETER

The command list FIFO is controlled by a dedicated block that takes the DE2D command and translates it into the required DMA configuration format for the three DMA channels used. The command interpreter executes commands that are fed from MIPS into the command FIFO. Execution of a command involves:

1. Programming DMA channels to read/write source/destination data from/to memory locations into DE2D read FIFOs
2. Triggering reads
3. Execution of the raster operations when DMA reads are completed
4. Writeback of the resultant data from write FIFO back to memory

This functionality can be controlled by programming the set of seven Operational Control registers that specifies the various Raster Operations (ROP) including the various rectangle sizes of the image on which the DE2D operates.

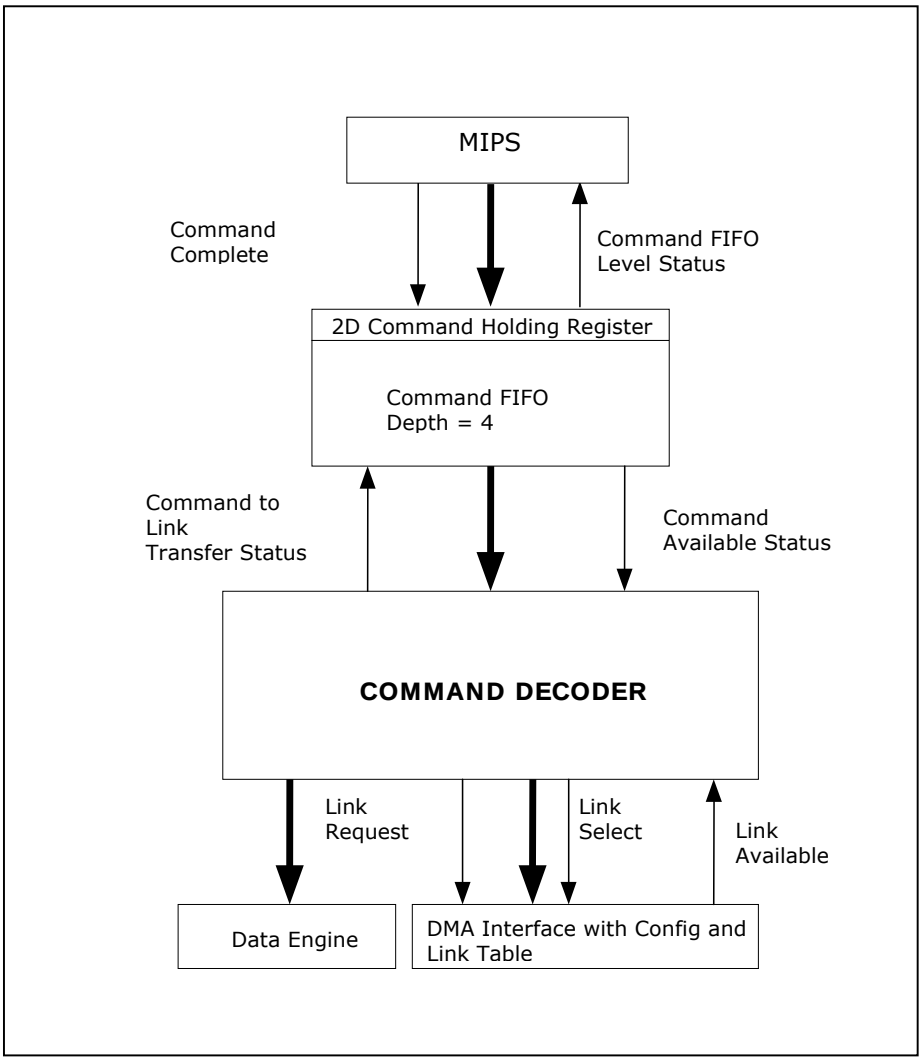


Figure 70. Command Interpreter Block Diagram

22.2 DATA ENGINE

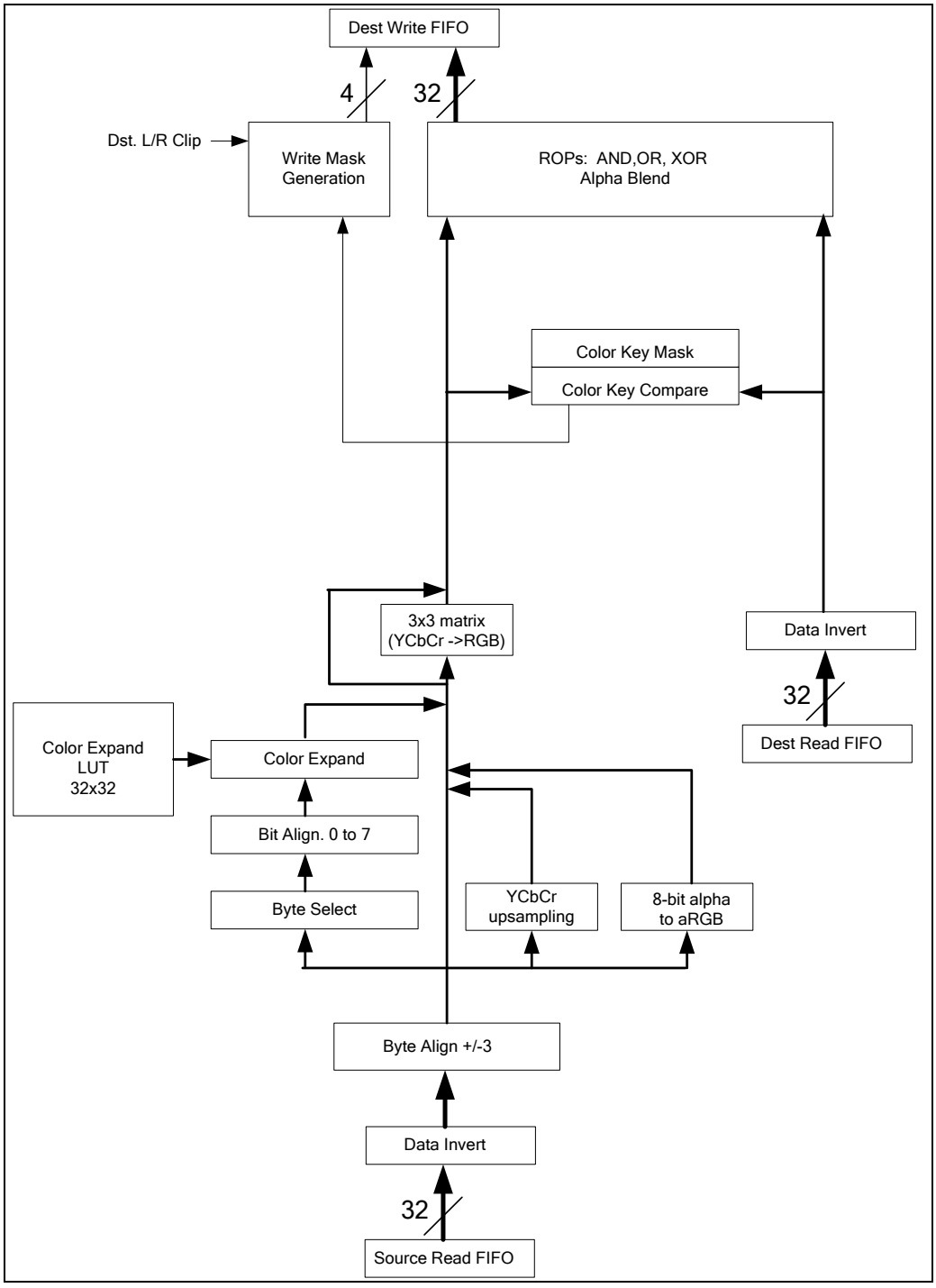


Figure 71. Data Engine Block Diagram

22.3 OPERATIONS SUPPORTED BY DE2D

The following operations are supported by the DE2D:

- 16 Microsoft defined logical Raster Operations on Source and Destination data
- Solid color fill of a rectangular memory area (this can also do horizontal and vertical lines)
- Color expansion of 1, 2, 4 and 8 bits/pixel Source data to the Display color depth
- Display pixel data alignment from Source data fetched at 32-bit data granularity
- Alpha blending of Source data into Destination data for 16 and 32bpp Destination formats
- Color-key function applied to Source data for transparency effects
- Color-key function applied to Destination data for write-protect effects
- 16-bit YCbCr 4:2:2 to 4:4:4 horizontal upsampling
- YCbCr ->aRGB color space conversion using programmable 3x3 matrix
- Hue/Saturation/Contrast/Brightness adjustment using programmable 3x3 matrix
- 8-bit alpha to aRGB conversion

22.4 PIXEL FORMATS SUPPORTED

Pixel formats supported are as follows:

- Display and Source data of 8 bits per pixel. The 8 bits are used as the index into a CLUT in the Graphics Display Processor data path.
- Display and Source data of 16 bits per pixel. Only the format of aRGB (1:5:5:5)
- Source aRGB (1:5:5:5) converted on the flight and blended with aRGB (8:8:8:8) Display data
- Display and Source data of 32 bits per pixel in format aRGB (8:8:8:8)
- Source data of YCbCr 4:2:2, color space converted to aRGB (8:8:8:8) or aRGB (1:5:5:5)
- Source data of 8-bit (alpha only, constant R, G and B) converted to aRGB (8:8:8:8) or aRGB (1:5:5:5)
- Source data of 1, 2, 4 or 8 bits per pixel to be color-expanded to the Display data format (8-bit index or 16-bit RGB or 32-bit RGB). These Source data formats use a 2D engine local color-expand look-up table.

The following table summarizes supported Source and Destination formats and their legal operations. An empty cell means no operations are supported on the =Source/Destination combination.

Table 45. Operations Supported on Different Source/Destination Formats

Destination Source	8-bit Index	16-bit 1:5:5:5	32-bit 8:8:8:8
1-bit CLUT	Color Expand, ROP	Color Expand, ROP Alpha blending	Color Expand, ROP Alpha blending
2-bit CLUT	Color Expand, ROP	Color Expand, ROP Alpha blending	Color Expand, ROP Alpha blending

Destination Source	8-bit Index	16-bit 1:5:5:5	32-bit 8:8:8:8
4-bit CLUT	Color Expand, ROP	Color Expand, ROP Alpha blending	Color Expand, ROP Alpha blending
8-bit CLUT	ROP	Color Expand, ROP Alpha blending	Color Expand, ROP Alpha blending
16-bit 1:5:5:5		ROP Alpha blending	ROP Alpha blending
32-bit 8:8:8:8			ROP Alpha blending
8-bit alpha		Alpha blending	Alpha blending
YCbCr16		Format conversion Alpha blending	Format conversion Alpha blending

The DE2D further supports the following image manipulation operations:

- Raster Operations (ROPs)
- Color Expansion
- Alpha Blend Mode
- Color Key
- Solid Color Fill

22.5 RASTER OPERATIONS

The DE2D supports two operand (Source and Destination) ROPs. The 16 logical operations listed below can be performed by controlling Source and Destination inversion and the logical operation between them (OR, AND, XOR).

Table 46. List of Raster Operations

MSB…LSB	Operation	Combination Rule
0000	0	All Zeros
0001	1	Source and Destination
0010	2	Source and NOT Destination
0011	3	Source
0100	4	NOT Source and Destination
0101	5	Destination
0110	6	Source XOR Destination
0111	7	Source OR Destination
1000	8	NOT Source AND NOT Destination
1001	9	NOT Source XOR Destination
1010	A	NOT Destination
1011	B	Source OR NOT Destination
1100	C	NOT Source
1101	D	NOT Source OR Destination
1110	E	NOT Source OR NOT Destination
1111	F	All Ones

22.6 COLOR EXPANSION

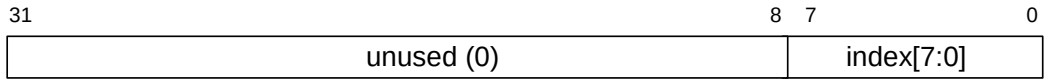
Color expansion of Source data to the Destination data format is supported for 1, 2, 4 or 8 bits/pixel. Source data is expanded through a 256 entry LUT with 32 bits per entry. 256 entries allow MIPS to change a block of 16 entries at the same time as the Drawing Engine is using the other 16 for 4bpp source.

If the Destination format is 16bpp or 32bpp, CLUT is organized as 256x32 LUT. The CLUT address is formed by 0, 4, 6 or 7 most-significant bits of COLOR_EXPAND_SELECT (at MSB position of the address) concatenated with 8, 4, 2 or 1 bits of pixel value (at LSB position of the address), depending on the Source format.

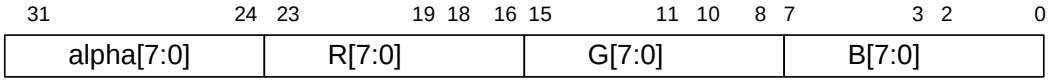
When the Destination format is 8bpp, the CLUT is organized as two LUTs, each with a size of 128x32. The Upper and lower part must be written by SW to the exactly identical value for all entries. 3, 5 or 7 most-significant bits of COLOR_EXPAND_SELECT are used in conjunction with 4, 2 or 1 bit of pixel value for CLUT indexing. Color expand of 8 bit Source to 8 bit Destination is not supported.

When the display format is 8bpp only the low byte of each LUT entry is used.

CLUT entry format when Destination is 8 bit is as follows:

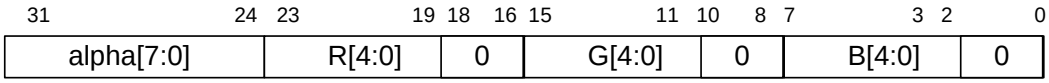


When the display format is 32bpp, all 32-bit of CLUT entries are used.



When the display format is 16bpp, the CLUT entry is organized identically to 32bpp mode, but 3 least-significant bits of each color component are unused. Unused bits in the CLUT must be written with '0'. Alpha is 8-bit, at [31:24], identical to 32bpp mode, but only the 5 most significant bits of alpha are used for alpha blending in 16bpp mode (bits 31:27). Correspondingly, 32 levels of alpha are available in 16bpp mode.

CLUT entry format when Destination is 16-bit is as follows:



Note: 8-bit alpha can be used for alpha blending of color-expanded source with 16bpp destination. However, if the selected operation is a ROP or source copy, only 1 least-significant bit of alpha is used – bit 24 of the CLUT entry. Similarly, if DESTINATION_ALPHA_WRITE is set to 010 (Source Alpha), only 1 least-significant bit of source alpha from CLUT will be used – bit 24 of the CLUT entry.

This allows for different source pixels to have different blend levels to offer better support for pre-rendered anti-aliased fonts. The text characters could be created as 2bpp format with four "colors"; opaque foreground, transparent background, and two level of translucent foreground as anti-aliased text edges.

22.6.1 ALPHA BLEND MODE

The 2D engine supports the following Alpha Composition modes summarized in the following table. C_S means any of R, G or B color components of source pixel, C_D means any of R, G or B color components of destination pixel, A_S and A_D mean alpha component of source and destination pixels. In all cases, destination alpha value is computed identically to C_D.

Table 47. Supported Alpha Composition Modes

Alpha Composition Mode	Formulae
Source Over	$C_D = C_S + C_D * (1- A_S)$
Destination Over	$C_D = C_S * (1- A_D) + C_D$
Source In	$C_D = C_S * A_D$
Destination In	$C_D = C_D * A_S$
Source Out	$C_D = C_S *(1- A_D)$
Destination Out	$C_D = C_D *(1- A_S)$
Source Atop	$C_D = C_S *A_D + C_D * (1- A_S)$
Destination Atop	$C_D = C_S * (1-A_D) + C_D * A_S$
Source XOR Destination	$C_D = C_S * (1-A_D) + C_D *(1- A_S)$

22.6.2 ALPHA BLENDING CONTROL

When the alpha value is 256 this equates to opaque, while an alpha value of 0 equates to transparent.

Field SOURCE_ALPHA_SEL of register SOURCE_PARAMETERS selects Source Alpha to be one of:

- Global Source Alpha from SOURCE_ALPHA_BLEND register
- Per-pixel Alpha from CLUT (if color-expand is enabled and source was 1/2/4/8bpp)
- Per-pixel Alpha (32-bpp only)

Similarly, field DESTINATION_ALPHA_SEL of register DESTINATION_PARAMETERS selects Destination Alpha as one of:

- Global Destination Alpha from DESTINATION_ALPHA_BLEND register
- Per-pixel Alpha (32-bpp only)

The alpha value in Source or Destination alpha register, CLUT, and per-pixel alpha in 32 bpp mode are in the range 0..255. Prior to being used in alpha blending, alpha values are remapped to the range 0..256 by incrementing by 1 all values >= 128 in 32bpp mode and all values >=16 in 16bpp mode.

For 32bpp Destination:

$$alpha\ used\ \ for\ blending[7:0] = \begin{cases} alpha, & if\ \ alpha < 128 \\ alpha + 1, & if\ \ alpha \geq 128 \end{cases}$$

Alpha value of 128 cannot be generated.

For 16bpp mode:

$$alpha\ used\ \ for\ blending[4:0] = \begin{cases} alpha, & if\ \ alpha < 16 \\ alpha + 1, & if\ \ alpha \geq 16 \end{cases}$$

SOURCE_ALPHA_MODE selects one of four possible alpha values as a multiplier for each Source Color components and Source Alpha:

- Source Alpha
- 256 - Source Alpha
- Destination Alpha
- 256 - Destination Alpha
- Value of 256
- Value of 0

DESTINATION_ALPHA_MODE has an identical control over Destination Color component and Alpha multiplier.

22.6.2.1 ALPHA BLENDING EXAMPLE

AWT Source Over blending is defined as:

$R_D = R_S + R_D * (1-A_S)$

$G_D = G_S + G_D * (1-A_S)$

$B_D = B_S + B_D * (1-A_S)$

$A_D = A_S + A_D * (1-A_S)$

For this blending SOURCE_ALPHA_SELECT would be set to '10' (per-pixel alpha), SOURCE_ALPHA_MODE set to '101' (fixed alpha = 256), DESTINATION_ALPHA_SELECT is 'don't care' and DESTINATION _ALPHA_MODE set to '000' (256-source alpha).

22.6.2.2 32-BPP ALPHA BLENDING

In 32bpp Destination Format, each color component and Source and Destination Alpha are 8 bit each. Bit 7 of the selected source alpha is added to the 8-bit alpha, resulting in a 9-bit alpha value in the range 0..256. *256 – alpha* is computed by inverting the selected alpha and adding '1' if the most significant bit of selected alpha was 0 ($a_S[7]==0$). The resulting value is 9-bit in the range 0..256.

- The multiplier is 9-bit x 8-bit and multiplication result is 16-bit
- The final adder is 17-bit

Destination Color and Alpha use bits [15:8] of the addition result. If bit [16] was set, the corresponding color component or alpha value is saturated to 0xFF and rounding is skipped.

Otherwise, rounding is performed: if bit [7] of addition result was set, a constant '1' is added to bits [15:8]. If carry is generated in this addition, the result is saturated to 0xFF.

22.6.2.3 16-BPP ALPHA BLENDING

For 16bpp destination, Source and Destination color components are 5 bit each, but the Alpha is 8 bit.

The same 8-bit multiplier as for 8-bit colors is used, 5-bit R, G and B values use bits [7:3] of multiplier input and bits [2:0] are cleared. Alpha uses all 8 bits, but alpha bits [2:0] all cleared to '0' in HW.

Range adjustment and 256-alpha implementation is different for 16-bpp case. If alpha bit 7 is set to'1', a value of '8' is added instead of '1' to 8-bit alpha.

256 – alpha is computed by inverting the selected alpha and adding '8' if the most significant bit of selected alpha was 0 ($a_S[7]==0$). The resulting value is 9-bit in the range 0..256.

As with 32-bpp, the multiplication result is 16-bit and final addition result has 17-bit.

Destination Color use bits [15:11] of addition result. If bit [16] was set, corresponding color component is saturated to 0x1F and rounding is skipped.

Otherwise, rounding is performed: if bit [10] of addition result was set, a constant '1' is added to bits [15:11]. If carry is generated in this addition, result is saturated to 0x1F.

Alpha Blending with 16bpp shows this scheme. Bit 16 affects saturation, marked 's'; bit 10 affects rounding, marked 'r'.

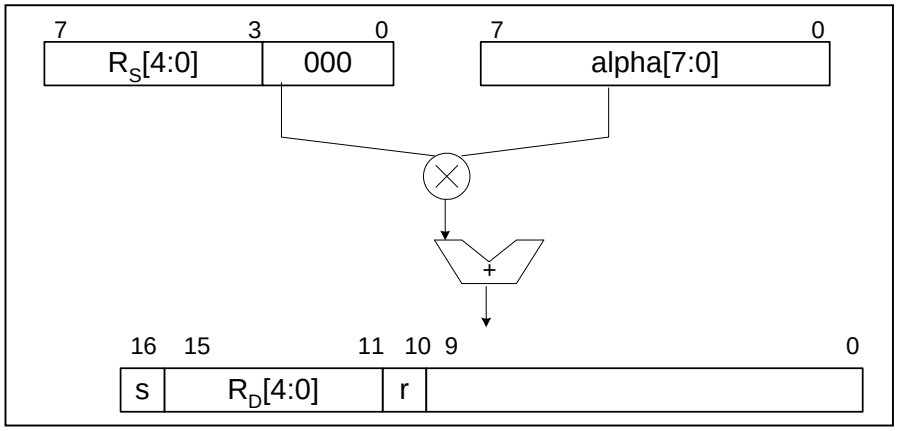


Figure 72. Alpha Blending with 16bpp

22.6.2.4 8-BIT ALPHA SOURCE FORMAT

Blit operation source of 8-bit alpha can be expanded to a:R:G:B 8:8:8:8 or 8:5:5:5 prior to blit using the DE2D_SOURCE_RGB_VALUE register. Which format is used is determined by the Destination format.

If the Destination format is 32bpp, each expanded pixel in a:R:G:B (8:8:8:8) format has alpha equal to source alpha and R, G, and B components equal to corresponding fields of the DE2D_SOURCE_RGB_VALUE register.

If the Destination format is 16-bit, 8-bit alpha pixel is expanded into a format identical to CLUT format in 16bpp. It has 8-bit alpha and 5-bit R (bits 23:19 of DE2D_SOURCE_RGB_VALUE register), G (bits 15:11) and B (bits 7:3) values. 8-bit alpha is used for alpha-blending with Destination pixels but is not stored to the Destination. (1-bit destination alpha is determined by field DESTINATION_ALPHA_WRITE in that case. If DESTINATION_ALPHA_WRITE is set to 010 (Source Alpha), only 1 least-significant bit of the Source alpha will be used.)

22.6.3 DESTINATIONS ALPHA CONTROL

For 16bpp Destination, alpha written to the Destination is controlled by field DESTINATION_ALPHA_WRITE of the register DESTINATION_PARAMETERS, $Alpha_D$ from alpha blending is ignored.

In 32bpp Destination, DESTINATION_ALPHA_WRITE selects between either preserving Source or Destination alpha, or setting or clearing all bits of 8-bit alpha. Two additional options are: Destination alpha written with $Alpha_D$ (Destination alpha result of alpha blending) or fixed alpha from register DESTINATION_ALPHA_VALUE.

22.7 COLOR KEY

Source or Destination data is compared to a color-key register value. The compare value has a bit-masking register associated to set "don't care" for any bit(s) in the compare. Destination data is from the output of the Destination data FIFO. Source data is after the color-expand function. A compare true may be used to protect (write-mask) Destination pixels or treat Source pixels as transparent. The Color Key and Mask are always 16 bit values. For 8 bit Display data, the software should program both bytes to the same value (this simplifies some hardware).

For example, there is a Source rectangle that contains an irregular object on a "background" of a color that is not part of the object. The object is to be copied to the display but the "background" is not. If the color key is set to the "background" color, only the pixels of the object will be written to the display.

22.8 SOLID COLOR FILL

The two DMA channels used for reads are kept idle. The Destination rectangle is filled with a single color defined in the Color Key register (the Color Key compare function is never used with a Fill).

22.9 YCBCR TO RGB FORMAT CONVERSION

The 2D engine can convert YCbCr 4:2:2 source pixels into aRGB 32-bit or 16-bit pixels during blit and then use the resulting 16/32-bit pixels for alpha blending with the Destination. The 2D engine treats YCbCr pixels as 32-bit pixels (2 4:2:2 pixels/32-bit).

The 2D Engine will interpolate 16-bit 4:2:2 pixel to 24-bit 4:4:4 and then format convert to RGB using a programmable 3x3 matrix. The Source alpha value from register DE2D_GLOBAL_PARAMETERS.SOURCE_ALPHA_BLEND will be filled in alpha value of each pixel.

Each color component of YCbCr source is 8-bit and the data layout is Cb0 Y0 Cr0 Y1 …

The SOURCE_FORMAT in register SOURCE_PARAMETERS must be set to '110' to select Cb Y Cr Y 4:2:2 format. This enables 4:2:2 to 4:4:4 upsampling.

22.9.1 4:2:2 TO 4:4:4 UPSAMPLING

Horizontal chroma upscaling (4:2:2 to 4:4:4) is performed by 2-tap interpolation (for odd i, $Cr_i=(Cr_{i-1}+Cr_{i+1})/2$; $Cb_i=(Cb_{i-1}+Cb_{i+1})/2$; for even i – Cr/Cb are not changed).

Since in 4:2:2 format every other Luma pixel has no corresponding Chroma information, it makes most the sense for each rectangle in YCbCr format to start on an even pixel boundary (i.e. 32-bit boundary) and to contain an even number of pixels. This would produce the most accurate color on right and left edges.

The 2D Engine only supports YCbCr input starting on an even pixel (i.e. the first byte contains Cb data) and having an even number of pixels. This aligns the start and end of YCbCr rectangle to 32-bit.

22.9.2 COLOR SPACE CONVERSION

The 2D Engine implements a 3x3 matrix with programmable coefficients. It is enabled by bit COLOR_MATRIX_EN of register DE2D_3X3_MATRIX_CONTROL, otherwise the 3x3 matrix is bypassed.

Input to this block can be YCbCr 4:4:4 data after upsampling, in which case YCbCr to RGB color space conversion is performed, or 16-bit or 32-bit RGB data for contrast/hue/saturation/brightness control.

The following matrix multiplication is performed:

$$\begin{bmatrix} G \\ B \\ R \end{bmatrix} = \begin{bmatrix} coef_{11} & coef_{12} & coef_{13} \\ coef_{21} & coef_{22} & coef_{23} \\ coef_{31} & coef_{32} & coef_{33} \end{bmatrix} \begin{bmatrix} Y \\ U/Cb \\ V/Cr \end{bmatrix} + \begin{bmatrix} off_{G/Y} \\ off_{B/U} \\ off_{R/V} \end{bmatrix}$$

Coefficients $coef_{11}$ to $coeff_{33}$ are 11-bit coefficients in 1 of 4 supported formats determined by MATRIX_COEF_FORMAT field of register DE2D_3X3_MATRIX_CONTROL. Coefficients can be programmed to comply with ITU-R. BT601 and with ITU-R. BT709 color systems.

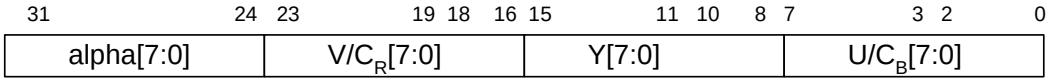
Luma values may be clipped to range 16..235 before applying the matrix. This is enabled by setting bit LUMA_CLIPPING_ENABLE in the register DE2D_YCBCR_CONVERSION.

Cb, Cr can be signed 2's complement numbers in the range -128..127 or positive offsets in the range 0..255. If Cb,Cr are signed numbers, bit CHROMA_SIGNED in register DE2D_YCBCR_CONVERSION must be set, if unsigned – cleared.

By changing matrix coefficients, contrast, hue and saturation can be controlled in addition to YCbCr to RGB conversion. Offsets vector is used for brightness adjustment.

Input to matrix multiplication can be YCbCr data after upsampling or aRGB 1:5:5:5 or 8:8:8:8. In case of aRGB input, alpha value is bypassed unmodified through this stage. If R, G and B were 5-bit, 3 zeros are appended on the right (least-significant bits) to each color component. After matrix multiplication the result would be converted back to 5-bit for R, G, and B.

The YCbCr input layout is as follows:



If Color Space Conversion is enabled, Y always maps to G, V/Cr always maps to R, and U/Cb always maps to B.

Output of matrix multiplication is RGB data in 8:8:8 format. It is then formatted to either aRGB 8:8:8:8 or 1:5:5:5. If the Destination format is 32bpp and the Source was YCbCr, the alpha of each resulting pixel is copied from SOURCE_ALPHA_BLEND field. Otherwise, (source was aRGB 8:8:8:8) the source pixel alpha is used unmodified.

If the Destination format is 16bpp and the Source was YCbCr, 3 least-significant bits of R, G and B are dropped and only 1 MSB of SOURCE_ALPHA_BLEND field is used for the resulting alpha. Otherwise (source was aRGB 1:5:5:5) the source pixel alpha is used unmodified.

23 GRAPHICS COMPOSITION

23.1 GRAPHICS DISPLAY PROCESSOR 1

Graphics Display Processor 1 (GDP1) receives graphics data from DMA and is usually used for native TV UI. The graphics data is usually in RGB formats.

The GDP processes either CLUT or non-CLUT input data, performs horizontal and vertical expansion, and transfers the data to the Graphics Blender to overlay the graphics data on the Main and PIP video.

The GDP1 internal pipeline is 32-bit and independent of input pixel format. In CLUT mode, input data from the graphics frame buffer is used as an index into the CLUT. The CLUT memory has 256 32-bit entries and the CLUT mode uses 8 bit input. The entries in the CLUT provide pixel values in 32-bit 8:8:8:8 format.

GDP1 supports two true color (non CLUT) formats: 16bpp and 32bpp. The 16bpp format is 1:5:5:5 - 1 bit of Alpha, 5 bits of Red, 5 bits of Green, and 5 bits of Blue. When alpha is zero, it is considered an opaque pixel. If alpha is one, it indicates that the alpha data is programmed by the blend register. 16-bit graphics data is expanded to 32-bit by appending 3 LSBs to each pixel value. The values of these bits are either specified by programmable registers or by copying 3 MSB values into the 3 LSBs. The 32 bpp format is 8:8:8:8.

GDP1 also supports a 3-tap Horizontal scaler with programmable coefficients and a 3-tap Vertical scaler with programmable coefficients. Horizontal and Vertical filters may be enabled independently – scalers are bypassed when graphics is generated in native (equal to display) resolution. Both scalers should only be used for expansion.

Maximum horizontal resolution which supports scaling is 1366 pixel/line. Any resolution up to 1080p is supported in scaler bypass mode.

23.1.1 COLOR SPACE CONVERSION

GDP1 implements a 3x3 matrix with programmable coefficients. This allows flexible color space conversion and contrast/hue/saturation/brightness control.

The following matrix multiplication is performed:

$$\begin{bmatrix} G \\ B \\ R \end{bmatrix} = \begin{bmatrix} coef_{11} & coef_{12} & coef_{13} \\ coef_{21} & coef_{22} & coef_{23} \\ coef_{31} & coef_{32} & coef_{33} \end{bmatrix} \begin{bmatrix} Y \\ U\ /\ Cb \\ V\ /\ Cr \end{bmatrix} + \begin{bmatrix} off_{G/Y} \\ off_{B/U} \\ off_{R/V} \end{bmatrix}$$

Coefficients $coef_{11}$ to $coeff_{33}$ are 11-bit coefficients in one of four supported formats. Coefficients can be programmed to comply with ITU-R. BT601 and with ITU-R. BT709 color systems.

GDP1 supports both alpha-premultiplied and not alpha-premultiplied graphics.

23.2 GRAPHICS DISPLAY PROCESSOR 2

Graphics Display Processor 2 (GDP2) is exactly identical to GDP1.

GDP2 may be used for graphics rendered at a resolution different from display, which needs an independent scaling factor from the TV UI. Examples include MHEG-5 or MHP graphics at standard definition.

23.3 GRAPHICS DISPLAY PROCESSOR 3

Graphics Display Processor 3 (GDP3) is similar to GDP1 but has no Vertical or Horizontal scaler. GDP3 graphics input resolution is always the same as the output resolution.

GDP3 may be used to generate subtitles or other graphics on VCR Output.

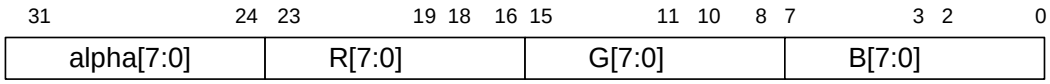
23.4 GRAPHICS BLENDER 1

Graphics Blender 1 blends the Main/PIP combined video plane with the physical Graphics Plane 1 (output of GDP1) and the physical Graphics Plane 2 (output of GDP2).

The Video Plane is after the Main and PIP Video Blender and also after Gamma LUT and dithering.

GDP1 and GDP2 active regions may completely or partially overlap.

Output of GDP1 and GDP2 is always 32bpp aRGB or aYUV 4:4:4, 8-bit each of R/V,G/Y,B/U, alpha in format:



Prior to alpha blending, alpha values are remapped to the range 0..256 by incrementing by 1 all values >= 128:

$$alpha\ used\ \ for\ blending[8:0] = \begin{cases} alpha, & if\ \ alpha < 128 \\ alpha + 1, & if\ \ alpha \geq 128 \end{cases}$$

The FLI106xx GFX Blender 1 supports alpha-premultiplied and not alpha-premultiplied graphics input, controlled by a register.

23.4.1 NOT ALPHA-PREMULTIPLIED GDP1 AND GDP2 GRAPHICS BLENDING

Normal FLI106xx video and graphics blender operation for Blend GDP1 with Video is as follows:

$R/G/B_{GDP1} * a_{GDP1} + R/G/B_{Video} * (1-a_{GDP1})$

$R/G/B_{GDP1}$ stands for R, G, and B components of GDP1 active pixel, a_{GDP1}stands for alpha of GDP1 pixel, $R/G/B_{Video}$ stands for corresponding component of Video pixel.

Note: outside of GDP1 active region $R/G/B/a_{GDP1} = 0$

Then blend GDP2:

$R/G/B_{GDP2} * a_{GDP2} + (R/G/B_{GDP1} * a_{GDP1} + R/G/B_{Video} * (1-a_{GDP1})) * (1-a_{GDP2}) =$

$R/G/B_{GDP2} * a_{GDP2} + R/G/B_{GDP1} * a_{GDP1} * (1-a_{GDP2}) + R/G/B_{Video} * (1-a_{GDP1}) * (1-a_{GDP2})$

23.4.2 ALPHA-PREMULTIPLIED GDP1 AND GDP2 GRAPHICS BLENDING

GDP1 blend result:

$R/G/B_{GDP1} + R/G/B_{Video} * (1-a_{GDP1})$

Then blend GDP2:

$R/G/B_{GDP2} + (R/G/B_{GDP1} + R/G/B_{Video} * (1-a_{GDP1})) * (1-a_{GDP2}) =$

$R/G/B_{GDP2} + R/G/B_{GDP1} * (1-a_{GDP2}) + R/G/B_{Video} * (1-a_{GDP1}) * (1-a_{GDP2})$

23.4.3 NOT ALPHA-PREMULTIPLIED GDP1 AND ALPHA-PREMULTIPLIED GDP2 BLENDING

$R/G/B_{GDP2} + R/G/B_{GDP1} * a_{GDP1} * (1-a_{GDP2}) + R/G/B_{Video} * (1-a_{GDP1}) * (1-a_{GDP2})$

23.4.4 ALPHA-PREMULTIPLIED GDP1 AND NOT ALPHA-PREMULTIPLIED GDP2 BLENDING

$R/G/B_{GDP2} * a_{GDP2} + R/G/B_{GDP1} * (1-a_{GDP2}) + R/G/B_{Video} * (1-a_{GDP1}) * (1-a_{GDP2})$

Note: The blending order is always as follows:

- Video is lowest layer
- GDP1 on top of Video
- GDP2 on top of GDP1 and Video

23.5 GRAPHICS BLENDER 2 – DEDICATED GRAPHICS OUTPUT OVER VXO

Graphics Blender 2 blends GDP1 and GDP2 graphics.

Combined graphics is output on VXO as 32bpp at ¼ of the video frame rate. Each GFX pixel is output over a rising and a falling edge of pixel clock of a 16-bit VXO interface, as shown in Figure 73.

Graphics Blender 2 has no Video input.

The result for not alpha-premultiplied GDP1 and GDP2 graphics:

$R/G/B_{GDP2} * a_{GDP2} + R/G/B_{GDP1} * a_{GDP1} * (1-a_{GDP2})$

The result for alpha-premultiplied GDP1 and GDP2 graphics:

$R/G/B_{GDP2} + R/G/B_{GDP1} * (1-a_{GDP2})$

Alpha value of $(1-a_{GDP1}) * (1-a_{GDP2})$ is output for alpha[7:0].

External Video Enhancement IC will then add unmodified graphics RGB values to video multiplied by the alpha received with the graphics.

External Video Enhancement IC alpha blending:

$R/G/B_{VXO} + R/G/B_{Video} * a_{VXO}$

Note: alpha in this case means Video transparency.

Graphics Blender 2 places GDP1 and GDP2 on a transparent background.

Outside of active GDP1 and GDP2 windows, R=G=B=0 and alpha[7:0] = 0xFF.

Graphics blender 2 frame rate is always ¼ of Main video frame rate.

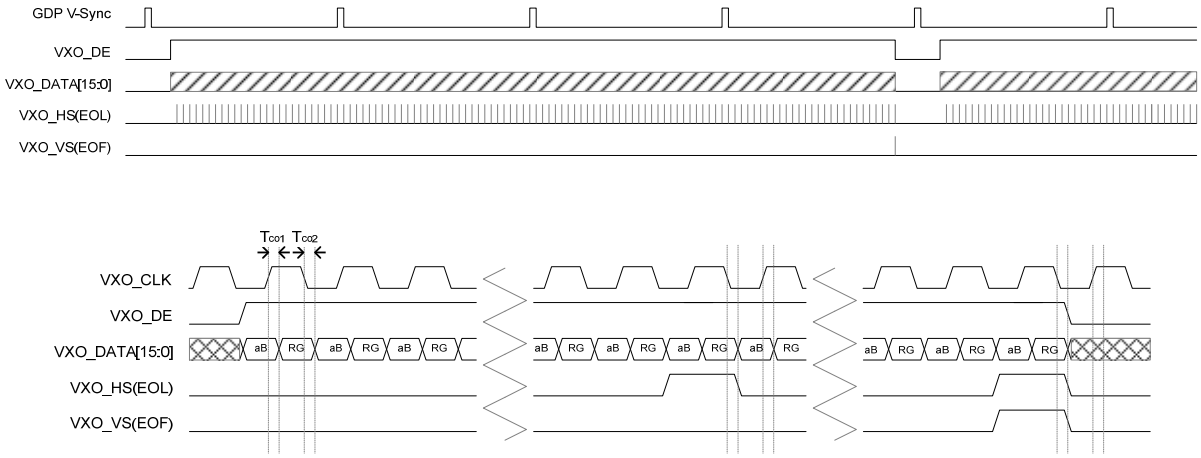


Figure 73. Graphic Blending Output Over VXO port in FLI106xx

23.6 GRAPHICS BLENDER 3

Graphics Blender 3 blends the output of GDP3 with video from the VCRO scalers, and feeds the PAL/NTSC Video Encoder.

Graphics Blender 3 is only used for DVB.

The maximum resolution for Graphics Plane 3 is 720x576p. GDP3 output is always interlaced (PAL).

Graphics Blender 3 operates in YCbCr 4:2:2 domain. If graphics were generated in RGB domain, it is converted to YCbCr by the 3x3 matrix then downsampled to 4:2:2 in GDP3.

24 VIDEO EXPANSION (VXI AND VXO PORTS)

The FLI106xx chip provides the ability to expand video processing capability using a Video Expansion Input (VXI) interface port and a Video Expansion Output (VXO) Interface port.

This expansion capability provides a scaleable solution for optional "add on" custom or proprietary video processing through external devices. This is done using a loop-back approach where any one of the FLI106xx input sources can be output through the VXO port, processed in an optional external IC, and provided back to the FLI106xx using the VXI port where it can then be either routed into the input of the main channel video processing pipeline, or alternatively, overlayed with the main channel video prior to graphics overlay and display output. This allows both the main channel video and any externally processed video to be overlaid with graphics from the powerful 2D engine (DE2D). The video loop-back technique avoids unnecessary graphics scaling and eliminates the requirement of a dedicated graphics channel between FLI106xx and backend video processing chips. The video expansion interface works with the Genesis family of chips (e.g. FLI8532) and also with third party video processing chips.

The VXI interface can also function as an additional input video stream, which can be selected as an input for the main or PIP channel video processing path within the FLI106xx. This allows the addition of other inputs, such as a DisplayPort receiver.

A third use for the VXI interface is that it can hook up to an external H.264 coprocessor. There is a serial TS out added in Douglas that shares pins with the VXI interface. However, in this mode, the classical VXI functionality is reduced to 16 pins. Thus, the VXI interface can be used as an input/output interface.

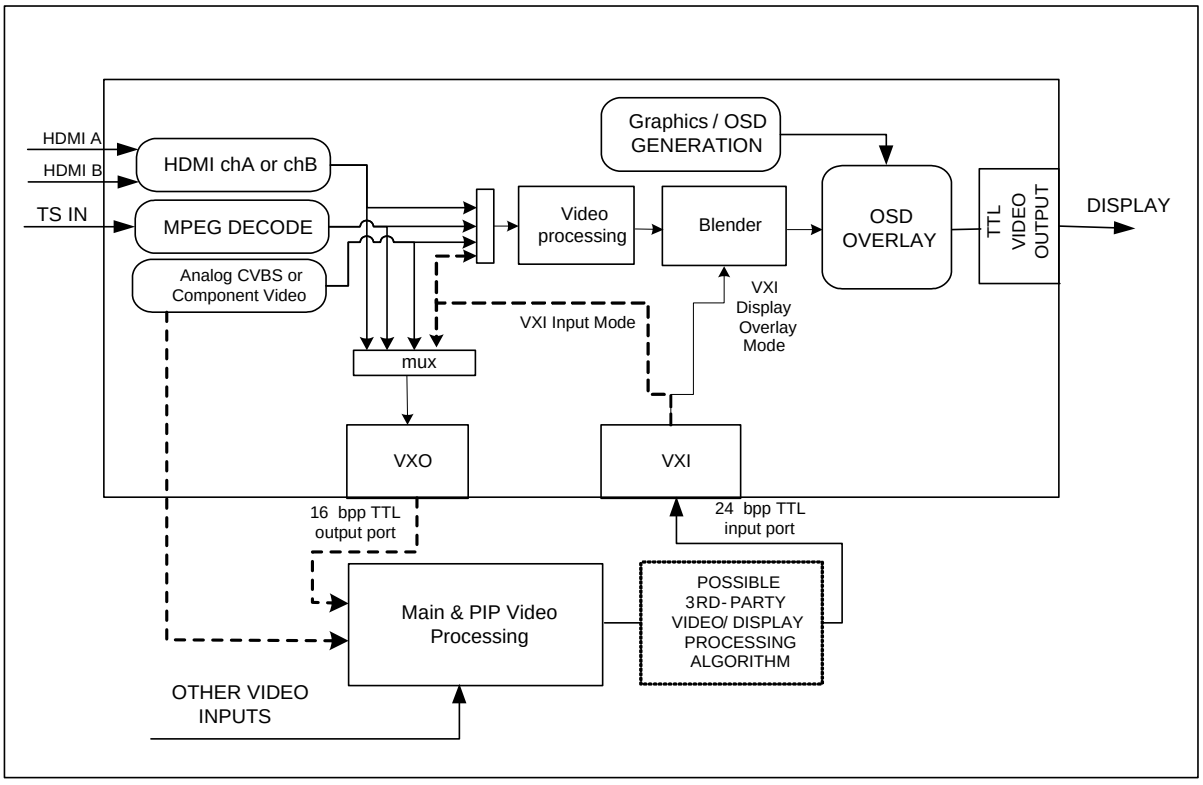


Figure 74. Video Loop Back Mechanism in FLI106xx

The block diagram in Video Loop Back Mechanism in FLI106xx demonstrates the video loop-back configuration and also shows ability of VXI as additional video input source.

Any one of the FLI106xx available video input streams can be selected for output on the VXO port. The VXO interface converts any input stream to 16 bit 4:2:2 format and outputs it at the pixel rate of selected input stream (video is not scaled to VXO port). RGB sources are converted to YCbCr. Maximum resolution supported is 1920 x 1080P and clock rates of 150 MHz max. The VXO port provides the pixel data, pixel clock, hsync, and vsync from the selected video source. The VXO port is connected to external display processor which can be a Genesis family of processor (e.g. FLI8532) or proprietary chip from CE integrator.

If the VXI input is intended to be blended with the FLI106xx main video channel, then both devices need to be frame locked and running at the same pixel clock frequency (with identical Htotal and Vtotal programming). FLI106xx can provide frame by frame synchronization to the external device or slave to the external device's Vsync or frame sync signal to synchronize on a frame by frame basis.

The VXI input can also be applied to the FLI106xx main or PIP video channel inputs for de-interlacing, scaling, frame rate conversion and color enhancement through the FLI106xx.

24.1 VXO PORT

Available Sources that can be selected for output on VXO Port:

- Source selected for PIP channel input

  This can be any of:

  - Decoded MPEG video (from MDTP)
  - SD Analog Video (from Analog Video Decoder)
  - Analog Component Video (or RGB Graphics from video input ADC's)
  - VXI (Video Expansion Input port)
  - HDMI Receiver
  - STG (Source Timing Generator for test purpose).
- Main video/graphics data after ACC-II, ACM-3D and Color Space Conversion
- Graphics from Graphics Blender 2. 32-bit per pixel graphics output at Double Data Rate (16-bit during falling VXO_CLK edge and 16-bit during rising VXO_CLK edge) at frame rate equal to ¼ of Main Video frame rate.
- 8-bit ITU-R BT.656 video from VCR Output block (SD only)
- 16-bit ITU-R BT.601 video from VCR Output block (SD only)

Pixel processing:

- Pass through YCbCr or YPbPr
- RGB to YCbCr conversion
- RGB to YCbCr conversion is programmable to support both Full range RGB (0-255) or EIA861 (16-235).
- 4:4:4 to 4:2:2 conversion
- Pass through RGB

VXO Control Pins: VXO_CLK, VXO_HS, VXO_VS, VXO_DE

Notes:

1) Tthe field id can optionally be output on the VXO_DE.

2) When the source is the Analog Component Video, then the VXO_DE signal is not available. The downstream processor should capture the desired pixel active video region for image processing.

VXO Data Pins: VXO_D[15:0]

Notes:

1) A single data rate (SDR) mode allows one pixel output per clock (16 bpp – all input data must be either originating as, or converted to YPbPr422 or YCbCr422.)

2) Optional DDR mode is supported which outputs ½ a pixel on both rising and falling edges of the clock. In DDR mode, then it is possible to output 24 bpp RGB data using 12 data pins or 16 bpp YCbCr or YPbPr 422 data using 8 data pins.

3) Control signals are always SDR changing state on the programmable rising or falling VXI_CLK edge.

Interfacing options:

- Programmable inversion for output control signals (ie. VXI_HS, VXI_VS, VXI_DE).
- Programmable Clock delay adjustment and Clock inversion (for VXI_CLK).
- Optional output of Field ID signal on the VXO_DEN pin.
- Programmable assignment of the pixel data (after color space and 444 to 422 conversion) onto output data pins. Assignment is done on a nibble basis, so any 4 bit nibble of selected video pixel can be programmed to appear on any group of 4 pins of the VXO data pins.
- In DDR mode, the programmable of assignment allows assigning any nibble of input pixel data to any 4 output pins rising edge, or any 4 output pins falling edge.
- Optional bit sequence reversal of data on VXO pins. (Example: VXO_D[7:4] could have the video source Y[3:0] assigned to those pins. By enabling nibble twist then pins VXO_D[7:4] could contain the data in reverse sequence: i.e., Y[0:3] data. This is an option to allow cleaner routing and shorter trace lengths on system PCB.
- GPIO – any pins not used for video output on the VXO port can be optionally programmed to function as GPIO pins.

By programming the Nibble swap here are examples of formats which can be supported:

pin	**16 bit SDR 422 YCbCr422 pixel0**	**16 bit SDR 422 YCbCr422 pixel1**	**12 bit DDR 888 RGB888 rising**	**12 bit DDR 888 RGB888 falling**	**12 bit DDR 888 YCbCr888 rising**	**12 bit DDR 888 YCbCr888 Falling**
VXO_D0	Y0	Y0	B0	G4	Cb0	Y4
VXO_D1	Y1	Y1	B1	G5	Cb1	Y5
VXO_D2	Y2	Y2	B2	G6	Cb2	Y6
VXO_D3	Y3	Y3	B3	G7	Cb3	Y7
VXO_D4	Y4	Y4	B4	R0	Cb4	Cr0
VXO_D5	Y5	Y5	B5	R1	Cb5	Cr1
VXO_D6	Y6	Y6	B6	R2	Cb6	Cr2
VXO_D7	Y7	Y7	B7	R3	Cb7	Cr3
VXO_D8	Cb0	Cr0	G0	R4	Y0	Cr4
VXO_D9	Cb1	Cr1	G1	R5	Y1	Cr5
VXO_D10	Cb2	Cr2	G2	R6	Y2	Cr6
VXO_D11	Cb3	Cr3	G3	R7	Y3	Cr7
VXO_D12	Cb4	Cr4	GPIO	GPIO	GPIO	GPIO
VXO_D13	Cb5	Cr5	GPIO	GPIO	GPIO	GPIO
VXO_D14	Cb6	Cr6	GPIO	GPIO	GPIO	GPIO
VXO_D15	Cb7	Cr7	GPIO	GPIO	GPIO	GPIO
HSYNC	HSYNC	HSYNC	HSYNC		HSYNC	
VSYNC	VSYNC	VSYNC	VSYNC		VSYNC	
DE	DE	DE	DE		DE	

pin	16 bit SDR 422 YCbCr422 pixel0	16 bit SDR 422 YCbCr422 pixel1	8 bit DDR 422 YCbCr422 pixl0 rising	8 bit DDR 422 YCbCr422 pixel0 falling	8 bit DDR 422 YCbCr422 pixel1 rising	8 bit DDR 422 YCbCr422 pixel1 falling	12 bit DDR 888 RGB rising	12 bit DDR 888 RGB falling
VXO_D0	Cb0	Cr0	Cb0	Y0	Cr0	Y0	B5	R1
VXO_D1	Cb1	Cr1	Cb1	Y1	Cr1	Y1	B6	R2
VXO_D2	Cb2	Cr2	Cb2	Y2	Cr2	Y2	B7	R3
VXO_D3	Cb3	Cr3	Cb3	Y3	Cr3	Y3	G0	R4
VXO_D4	Cb4	Cr4	Cb4	Y4	Cr4	Y4	G1	R5
VXO_D5	Cb5	Cr5	Cb5	Y5	Cr5	Y5	G2	R6
VXO_D6	Cb6	Cr6	Cb6	Y6	Cr6	Y6	G3	R7
VXO_D7	Cb7	Cr7	Cb7	Y7	Cr7	Y7		
VXO_D8	Y0	Y0	GPIO	GPIO	GPIO	GPIO	GPIO	GPIO
VXO_D9	Y1	Y1	GPIO	GPIO	GPIO	GPIO	GPIO	GPIO
VXO_D10	Y2	Y2	GPIO	GPIO	GPIO	GPIO	GPIO	GPIO
VXO_D11	Y3	Y3	GPIO	GPIO	GPIO	GPIO	GPIO	GPIO
VXO_D12	Y4	Y4	GPIO	GPIO	GPIO	GPIO	B0	G4
VXO_D13	Y5	Y5	GPIO	GPIO	GPIO	GPIO	B1	G5
VXO_D14	Y6	Y6	GPIO	GPIO	GPIO	GPIO	B2	G6
VXO_D15	Y7	Y7	GPIO	GPIO	GPIO	GPIO	B3	G7
HSYNC	HSYNC	HSYNC	HSYNC		HSYNC		HSYNC	
VSYNC	VSYNC	VSYNC	VSYNC		VSYNC		VSYNC	
DE	DE	DE	DE		DE		DE	

pin	16 bit SDR ITU-R BT.601	8 bit SDR ITU-R BT.656	10 bit DDR 422 10-bit YCbCr422 rising	10 bit DDR 422 10-bit YCbCr422 falling	16 bit DDR 444 10-bit RGB rising	16 bit DDR 444 10-bit RGB falling	16 bit DDR 32-bit a:R:G:B Rising	16 bit DDR 32-bit a:R:G:B RGB falling
VXO_D0	601_C0	GPIO	0	0	R0	0	R0	B0
VXO_D1	601_C1	GPIO	0	0	R1	0	R1	B1
VXO_D2	601_C2	GPIO	Cb0/Cr0	Y0	B0	G0	R2	B2
VXO_D3	601_C3	GPIO	Cb1/Cr1	Y1	B1	G1	R3	B3
VXO_D4	601_C4	GPIO	Cb1/Cr2	Y2	B2	G2	R4	B4
VXO_D5	601_C5	GPIO	Cb1/Cr3	Y3	B3	G3	R5	B5
VXO_D6	601_C6	GPIO	Cb1/Cr4	Y4	B4	G4	R6	B6
VXO_D7	601_C7	GPIO	Cb1/Cr5	Y5	B5	G5	R7	B7
VXO_D8	601_y0	656_D0	Cb1/Cr6	Y6	B6	G6	G0	α0
VXO_D9	601_y1	656_D1	Cb1/Cr7	Y7	B7	G7	G1	α1
VXO_D10	601_y2	656_D2	Cb1/Cr8	Y8	B8	G8	G2	α2
VXO_D11	601_y3	656_D3	Cb1/Cr9	Y9	B9	G9	G3	α3
VXO_D12	601_y4	656_D4	GPIO	GPIO	R2	R6	G4	α4
VXO_D13	601_y5	656_D5	GPIO	GPIO	R3	R7	G5	α5
VXO_D14	601_y6	656_D6	GPIO	GPIO	R4	R8	G6	α6
VXO_D15	601_y7	656_D7	GPIO	GPIO	R5	R9	G7	α7

HSYNC	601_HSYNC	HSYNC	HSYNC	HSYNC
VSYNC	601_VSYNC	VSYNC	VSYNC	VSYNC
DE	601_DE	DE	DE	DE

20-bit YCbCr or 30-bit RGB can only be output in DDR mode as shown above.

VXO can also be used to output 32-bit a:R:G:B graphics, typically used with external MEMC. This graphics is generated by Graphics Blender 2 by blending Graphics Plane 1 (GDP1) and, optionally, Graphics Plane 2 (GDP2).

656/601 output can be Main or PIP video downscaled to SD resolution within VCR Output block.

24.2 VXI PORT (DIGITAL INPUT PORT)

The VXI Port operates as a Digital TTL Input Port to supply video input for two primary modes:

1. 24-bit TTL input to Main or PIP processing pipeline
2. Display Overlay (VXI must be synchronized with display timing for PIP or Main video overlay)

It also operates in an input/output mode with the 5 MSB used as TS serial out and the bottom 16 LSB used as uncompressed video in. The Cx24182 can connect to VXI interface only in this input/output configuration leaving VXO available for other video enhancement external chips, such as MCTi from Genesis, or other MEMC chips.

The FLI106xx supports a 24-bit digital input port with digital sync inputs, field indicator, data enable, and two clock inputs. The 24-bit input can be flexibly configured to support a wide range of digital sources and formats.

Inputs to the digital input port are TTL-compatible with a maximum clock speed of 150 MHz. Sync and clock polarities are programmable.

24.2.1.1 SUPPORTED DIGITAL INPUT FORMATS

The following digital video formats are supported by the FLI106xx digital video/graphic port:

- 8/10-bit ITU-BT.656
- 16/20-bit ITU-BT-601
- 16/20-bit SMPTE274 (1080i)
- 16/20-bit SMPTE296 (720p)
- 8-bit 4:2:2 YCbCr or YPbPr
- 16-bit 4:2:2 YCbCr or YPbPr
- 20-bit 4:2:2 YCbCr or YPbPr
- 24-bit 4:4:4 YCbCr or YPbPr
- 30-bit 4:4:4 YCbCr or YPbPr (DDR mode only)
- 24-bit RGB
- 30-bit RGB (DDR mode only)
- 16-bit 4:4:4 YCbCr input + 8-bit Serial Transport Stream Output

The following figures illustrate the timing of the video formats:

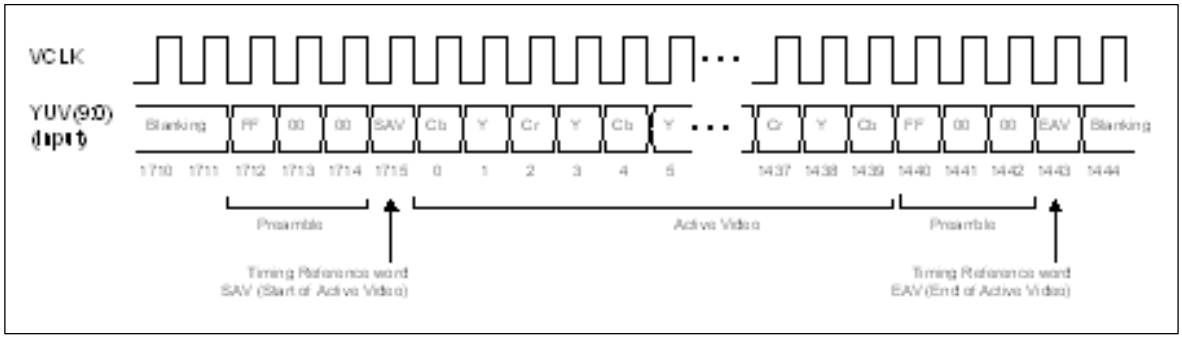


Figure 75. ITU-R BT656 Input

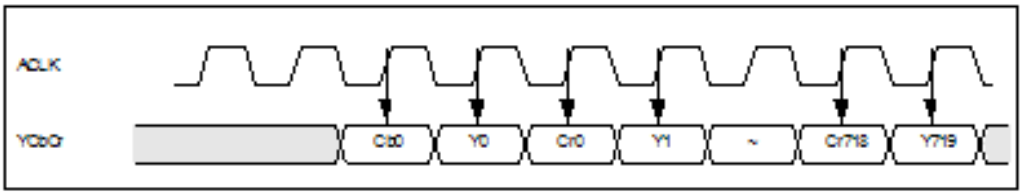


Figure 76. 8-bit 4:2:2 YCbCr/YPbPr

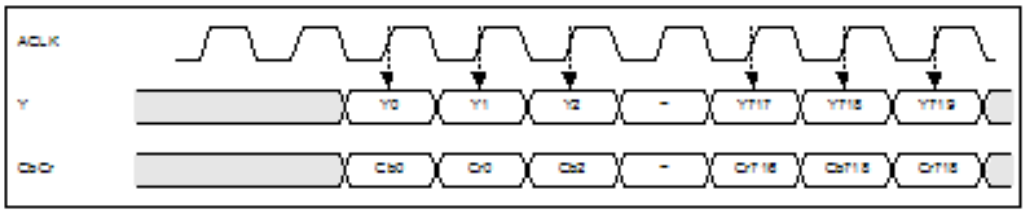


Figure 77. 16/20-bit 4:2:2 YCbCr/YPbPr

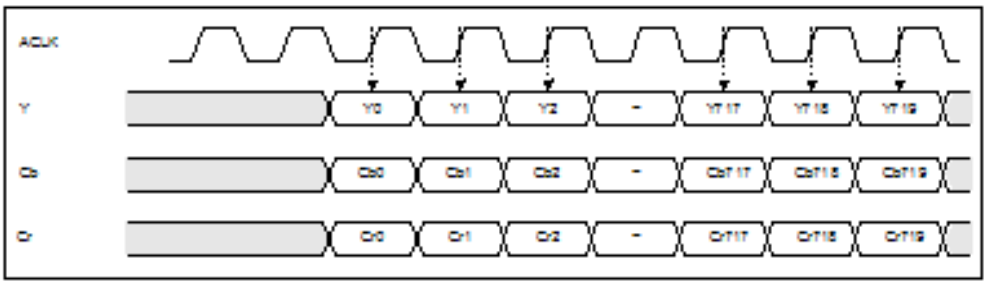


Figure 78. 24-bit 4:4:4 YCbCr/YPbPr (SDR)

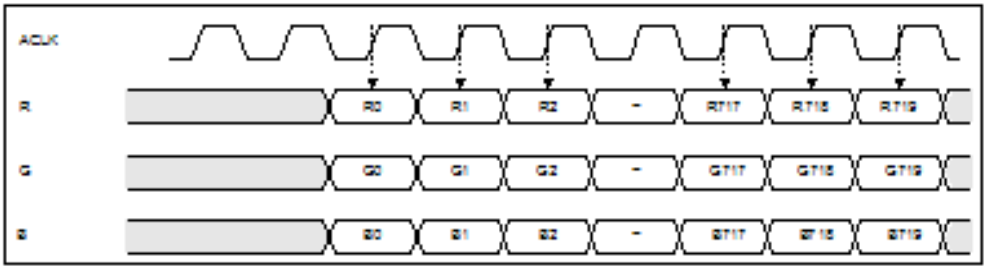


Figure 79. 24-bit RGB (SDR)

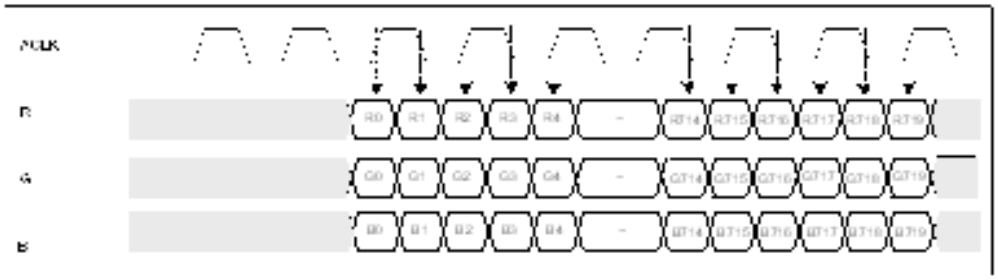


Figure 80. 30-bit DDR (Double Data Rate) Data Transfer

24.2.2 656 DECODER

The ITU-BT-656 video format consists of pixel clock and 8 bits/10 bits of data depending on the input. No separate HSync, VSync, and odd signals are present. Timing data is embedded in the data stream. The internal 656 decoder will extract the HSync, VSync, and odd signals from the embedded timing data.

Decoder also extracts the ancillary data embedded into 656/274/296 video streams.

24.2.3 VXI INPUT MODE

The VXI port input video may be selected as a source to either Main or PIP processing pipeline. This allows the VXI port to be used as a generic digital video input port for connecting inputs from sources such as a DisplayPort Receiver, external HDMI receiver or H.264 decoder IC.

24.2.4 VXI DISPLAY (OR PIP) OVERLAY

The VXI port input video is passed through a small FIFO and input to the PIP blender for merging with the on-chip main video processing channel. The VXI port video is merged with the main channel video prior to the graphics blender so that the graphics/OSD overlay is not affected by either of the video (Main channel or VXI) video streams.

When using VXI Display Overlay mode, the external VXI video source (typically an external PIP processing chip or video enhancement chip) both must be pixel locked, line locked, and frame locked so that real time overlay can be achieved at display resolution. The FIFO in the data path resolves minor (a few display clocks) deviation between the VXI source timing and the FLI106xx display timing so synchronization between the two is needed.

The FLI106xx can lock to an external frame-by-frame synchronization signal (DFSYNC pin or VXI_VS) and also lock its display pixel clock to the VXI_CLK input pin. Alternatively, the FLI106xx can output a once-per-display frame synchronization signal to the external video processor, which is supplying the video input to the VXI interface.

For VXI Display Overlay mode, the following are supported:

Sampling Modes: 4:4:4

Color Space Modes: VXI input must be in the same color space as the FLI106xx display output. (The input data from VXI interface is directly merged with the main channel video stream. There is no color space conversion in the VXI path for Display Overlay Mode. The external PIP or video enhancement processor supplying video for display overlay must convert the video stream to the color space of the display device.)

VXI Control Pins: VXI_CLK, VXI_HS, VXI_VS, VXI_DE

Note: the field id can optionally be output on the VXI_DE.

VXI Data Pins: VXI_D[23:0]

Note: a single data rate (SDR) mode allows one pixel per clock.

Optional DDR mode is supported which inputs ½ a pixel on both rising and falling edges of the clock. In DDR mode, then it is possible to input 24 bpp RGB data using 12 data pins or 16 bpp YCbCr 422 data using 8 data pins.

Interfacing options:

- Programmable inversion for intput control signals (ie. VXI_HS, VXI_VS, VXI_DE).
- Programmable Clock delay adjustment and Clock inversion (for VXI_CLK).
- Programmable assignment of data pins to the internal pixel bus. Assignment is done on a nibble basis, so any group of 4 VXI data pins can be assigned to any 4 bits of the 24 bit internal pixel bus.
- In DDR mode, independent assignment of the input data sampled on rising or falling edges is supported.
- Optional bit sequence reversal of input data on VXI pins. (Example: VXI_D[7:4] could have the video source Y[3:0] assigned to those pins. By enabling nibble twist, then pins VXI_D[7:4] could contain the data in reverse sequence: i.e., Y[0:3] data. This is an option to allow cleaner routing and shorter trace lengths on system PCB.

24.2.4.1 TRANSPORT STREAM OUTPUT MODE

VXI Data bits VXI_D[23] to VXI_D[18] may be used as Serial Transport Stream output.

VXI_D[23] = TS_CLK(o),

VXI_D[22] = TS_D(o),

VXI_D[21] = TS_SYNC(o),

VXI_D[20] = TS_VALID(o),

VXI_D[19] = DREQ(I)

VXI_D[18] =TS_ERROR (o)

Bits Bits VXI_D[17:16] may be used as GPIO. Bits DREQ(I) and TS_ERROR (o) are optional, if not used for TS Out then VXI_D[19:18] may be used as GPIO.

Bits VXI_D[15:0], VXI_CLK. VXI_VS, VXI_HS, VXI_DE retains normal VXI functionality in this mode, but data is limited to 16 bits.

Examples of input connection formats which can be supported by programming the pin assignment are shown below:

pin	Si 24 bit SDR 888 RGB888	Si 24 bit SDR 888 YCbCr888	Si 16 bit SDR 422 YCbCr422 pixel0	Si 16 bit SDR 422 YCbCr422 pixel1	Si 12 bit DDR 888 RGB888 rising	Si 12 bit DDR 888 RGB888 falling	Si 12 bit DDR 888 YCbCr888 rising	Si 12 bit DDR 888 YCbCr888 falling	AD 24 bit SDR 888 RGB888	AD 24 bit SDR 888 YCbCr888
VXI_D0	B0	Cb0			B0	G4	Cb0	Y4	B0	Cb0
VXI_D1	B1	Cb1			B1	G5	Cb1	Y5	B1	Cb1
VXI_D2	B2	Cb2			B2	G6	Cb2	Y6	B2	Cb2
VXI_D3	B3	Cb3			B3	G7	Cb3	Y7	B3	Cb3
VXI_D4	B4	Cb4			B4	R0	Cb4	Cr0	B4	Cb4
VXI_D5	B5	Cb5			B5	R1	Cb5	Cr1	B5	Cb5
VXI_D6	B6	Cb6			B6	R2	Cb6	Cr2	B6	Cb6
VXI_D7	B7	Cb7			B7	R3	Cb7	Cr3	B7	Cb7
VXI_D8	G0	Y0	Cb0	Cr0	G0	R4	Y0	Cr4	G0	Y0
VXI_D9	G1	Y1	Cb1	Cr1	G1	R5	Y1	Cr5	G1	Y1
VXI_D10	G2	Y2	Cb2	Cr2	G2	R6	Y2	Cr6	G2	Y2
VXI_D11	G3	Y3	Cb3	Cr3	G3	R7	Y3	Cr7	G3	Y3
VXI_D12	G4	Y4	Cb4	Cr4					G4	Y4
VXI_D13	G5	Y5	Cb5	Cr5					G5	Y5
VXI_D14	G6	Y6	Cb6	Cr6					G6	Y6
VXI_D15	G7	Y7	Cb7	Cr7					G7	Y7
VXI_D16	R0	Cr0	Y0	Y0					R0	Cr0
VXI_D17	R1	Cr1	Y1	Y1					R1	Cr1
VXI_D18	R2	Cr2	Y2	Y2					R2	Cr2
VXI_D19	R3	Cr3	Y3	Y3					R3	Cr3
VXI_D20	R4	Cr4	Y4	Y4					R4	Cr4
VXI_D21	R5	Cr5	Y5	Y5					R5	Cr5
VXI_D22	R6	Cr6	Y6	Y6					R6	Cr6
VXI_D23	R7	Cr7	Y7	Y7					R7	Cr7
VXI_HS	HSYNC	HSYNC	HSYNC	HSYNC	HSYNC		HSYNC		HSYNC	HSYNC
VXI_VS	VSYNC	VSYNC	VSYNC	VSYNC	VSYNC		VSYNC		VSYNC	VSYNC
VXI_DE	DE	DE	DE	DE	DE		DE		DE	DE

pin	AD 16 bit SDR 422 YCbCr422 pixel0	AD 16 bit SDR 422 YCbCr422 pixel1	AD 8 bit DDR 422 YCbCr422 pixl0 rising	AD 8 bit DDR 422 YCbCr422 pixel0 falling	AD 8 bit DDR 422 YCbCr422 pixel1 rising	AD 8 bit DDR 422 YCbCr422 pixel1 falling	AD 12 bit DDR 888 RGB rising	AD 12 bit DDR 888 RGB falling
VXI_D0			Cb0	Y0	Cr0	Y0		
VXI_D1			Cb1	Y1	Cr1	Y1		
VXI_D2			Cb2	Y2	Cr2	Y2		
VXI_D3			Cb3	Y3	Cr3	Y3		
VXI_D4			Cb4	Y4	Cr4	Y4		
VXI_D5			Cb5	Y5	Cr5	Y5		
VXI_D6			Cb6	Y6	Cr6	Y6		
VXI_D7			Cb7	Y7	Cr7	Y7		
VXI_D8	Y0	Y0						
VXI_D9	Y1	Y1						
VXI_D10	Y2	Y2						

VXI_D11	Y3	Y3				
VXI_D12	Y4	Y4			B0	G4
VXI_D13	Y5	Y5			B1	G5
VXI_D14	Y6	Y6			B2	G6
VXI_D15	Y7	Y7			B3	G7
VXI_D16	Cb0	Cr0			B4	R0
VXI_D17	Cb1	Cr1			B5	R1
VXI_D18	Cb2	Cr2			B6	R2
VXI_D19	Cb3	Cr3			B7	R3
VXI_D20	Cb4	Cr4			G0	R4
VXI_D21	Cb5	Cr5			G1	R5
VXI_D22	Cb6	Cr6			G2	R6
VXI_D23	Cb7	Cr7			G3	R7
VXI_HS	HSYNC	HSYNC	HSYNC	HSYNC	HSYNC	
VXI_VS	VSYNC	VSYNC	VSYNC	VSYNC	VSYNC	
VXI_DE	DE	DE	DE	DE	DE	

	AD	AD	AD	AD	AD	AD
pin	12 bit DDR	12 bit DDR	12 bit DDR	12 bit DDR	24 bit SDR	24 bit SDR
	422	422	422	422	422	422
	YCbCr	YCbCr	YCbCr	YCbCr	YCbCr	YCbCr
	rising	falling	rising	falling	pixel0	pixel1
VXI_D0	Cb0	Y0	Cr0	Y0	Y0	Y0
VXI_D1	Cb1	Y1	Cr1	Y1	Y1	Y1
VXI_D2	Cb2	Y2	Cr2	Y2	Y2	Y2
VXI_D3	Cb3	Y3	Cr3	Y3	Y3	Y3
VXI_D4	Cb4	Y4	Cr4	Y4	Y4	Y4
VXI_D5	Cb5	Y5	Cr5	Y5	Y5	Y5
VXI_D6	Cb6	Y6	Cr6	Y6	Y6	Y6
VXI_D7	Cb7	Y7	Cr7	Y7	Y7	Y7
VXI_D8	Cb8	Y8	Cr8	Y8	Y8	Y8
VXI_D9	Cb9	Y9	Cr9	Y9	Y9	Y9
VXI_D10	Cb10	Y10	Cr10	Y10	Y10	Y10
VXI_D11	Cb11	Y11	Cr11	Y11	Y11	Y11
VXI_D12					Cb0	Cr0
VXI_D13					Cb1	Cr1
VXI_D14					Cb2	Cr2
VXI_D15					Cb3	Cr3
VXI_D16					Cb4	Cr4
VXI_D17					Cb5	Cr5
VXI_D18					Cb6	Cr6
VXI_D19					Cb7	Cr7
VXI_D20					Cb8	Cr8
VXI_D21					Cb9	Cr9
VXI_D22					Cb10	Cr10
VXI_D23					Cb11	Cr11
VXI_HS	HSYNC		HSYNC		HSYNC	HSYNC
VXI_VS	VSYNC		VSYNC		VSYNC	VSYNC
VXI_DE	DE		DE		DE	DE

	AD	AD	AD
pin	20 bit SDR	20 bit SDR	24 bit SDR
	422		16-bit 422 YCbCr + TS Out
	YCbCr	SMPTE274	
VXI_D0			Cb0/Cr0
VXI_D1			Cb1/Cr1
VXI_D2	Cb0/Cr0	Cb0/Cr0	Cb2/Cr2
VXI_D3	Cb1/Cr1	Cb1/Cr1	Cb3/Cr3
VXI_D4	Cb2/Cr2	Cb2/Cr2	Cb4/Cr4
VXI_D5	Cb3/Cr3	Cb3/Cr3	Cb5/Cr5
VXI_D6	Cb4/Cr4	Cb4/Cr4	Cb6/Cr6
VXI_D7	Cb5/Cr5	Cb5/Cr5	Cb7/Cr7
VXI_D8	Cb6/Cr6	Cb6/Cr6	Y0
VXI_D9	Cb7/Cr7	Cb7/Cr7	Y1
VXI_D10	Cb8/Cr8	Cb8/Cr8	Y2
VXI_D11	Cb9/Cr9	Cb9/Cr9	Y3
VXI_D12			Y4
VXI_D13			Y5
VXI_D14	Y0	Y0	Y6
VXI_D15	Y1	Y1	Y7
VXI_D16	Y2	Y2	GPIO
VXI_D17	Y3	Y3	GPIO
VXI_D18	Y4	Y4	TS_ERROR (o)
VXI_D19	Y5	Y5	DREQ(I)
VXI_D20	Y6	Y6	TS_VALID(o)
VXI_D21	Y7	Y7	TS_SYNC(o)
VXI_D22	Y8	Y8	TS_D(o)
VXI_D23	Y9	Y9	TS_CLK(o)
VXI_HS	HSYNC		HSYNC
VXI_VS	VSYNC		VSYNC
VXI_DE	DE		DE

25 DISPLAY OUTPUT INTERFACE

The Display Output Port provides data and control signals that permit the FLI106xx to connect to a variety of display devices using a TTL or LVDS interface. The output interface is configurable for LVDS in 18, 24, or 30-bit RGB pixel format in single- or double-wide formats. TTL output is available in 18/24/30-bit RGB pixel format in single-wide output mode as well as 20 and 24/30 4:2:2 YUV single-wide format.

All display data and timing signals are synchronous with the DCLK output clock. The integrated LVDS transmitter is programmable to allow the data and control signals to be re-mapped to support all common LVDS receiver formats. DC balanced operation is supported as described in the Open LDI standard.

Note: If the output is 4:4:4, the width can be 30 bits (10 bits per channel) or 24 bits (8 bits per channel) and YUV is re-mappable through the 3x3 matrix. If the output is 4:2:2, the 3x3 matrix can only be used for color space conversion and there are three fixed options:

a. 4:2:2, 8 bits per channel, outputs allocation:

   G = Y, B = UV

b. 4:2:2, 10 bits per channel, outputs allocation:

   G = Y, B = UV

c. 4:2:2, 12 bits per channel, outputs allocation:

   G[7:0] = Y[11:4], B[7:0] = UV[11:4], R[3:0] = UV[3:0], R[7:4] = Y [3:0]

Note: FLI106xx supports the ability to program the display background color in RGB or YUV (5:6:5) formats.

25.1 DISPLAY SYNCHRONIZATION

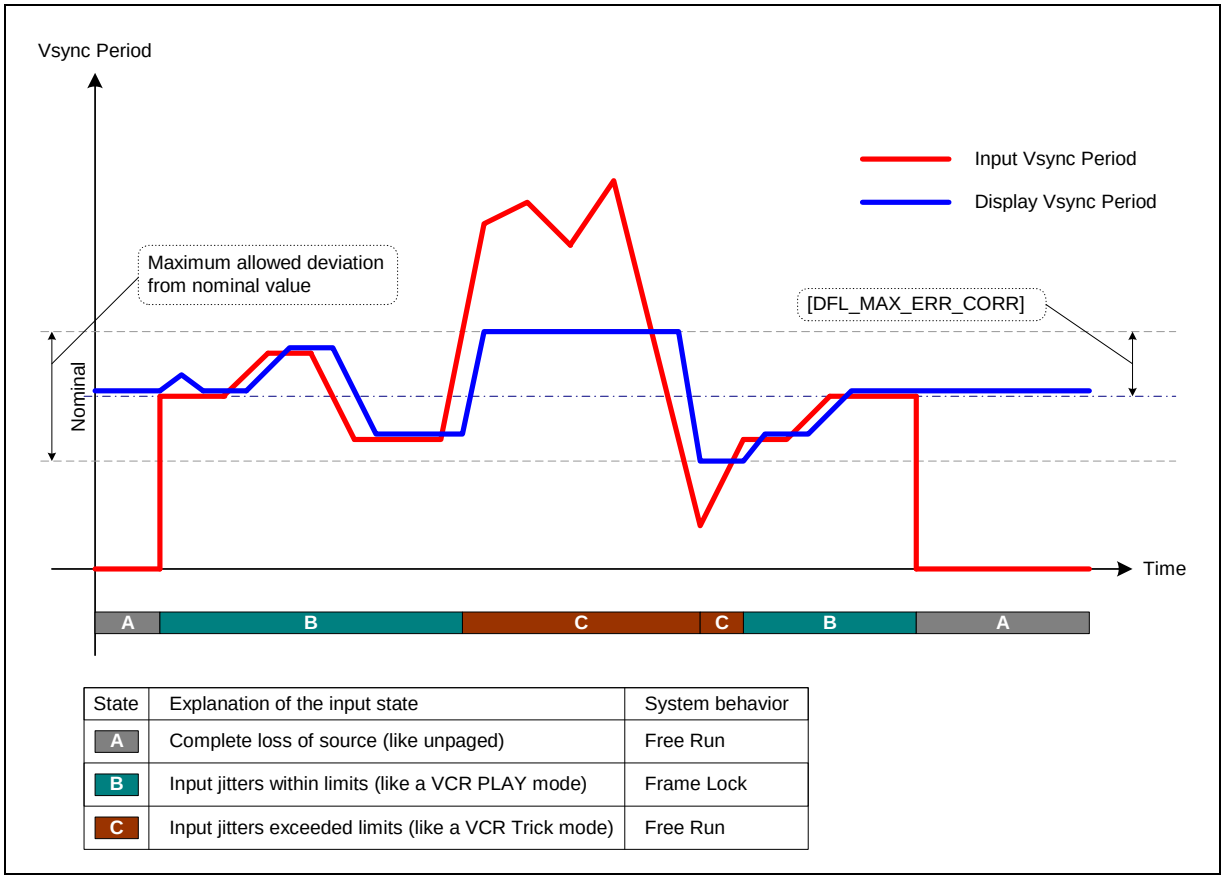
The FLI106xx display synchronization modes are:

- **Frame Sync Mode:** The display frame rate is synchronized to the input frame or field rate. This mode is used for standard operation. Frame rate conversion is optionally performed. The display frame can be locked to the Main or the PIP channel.
- **Free Run Mode:** (No synchronization) This mode is used when there is no valid input timing (i.e. to display OSD messages or a splash screen) or for testing purposes. In free-run mode, the display timing is determined only by the values programmed into the display window and timing registers.
- **DFL Mode:** (Dynamic Frame Lock) The goal of this mode is to slowly lock the display to the input whenever the difference between the input and output timing is smaller than the maximum threshold [DFL_MAX_ERR_CORR], otherwise the synchronizer just free runs. In order to achieve this goal, the block will perform two types of measurement. First, it measures the difference between input and output period. Second the input-output phase difference. After taking the measurements, compare between the period error and the maximum allowed correction (in pixels) [DFL_MAX_ERR_CORR]. Based on this comparison do one of the following:

  a. When the period error is smaller than the maximum error correction, which means that the input-output timing difference is small (achievable), the Display Synchronizer will start adding/dropping pixels and/or lines in a predefined locations using the measured phase error (adding/dropping pixels can be done per line in the vertical front porch or in every line). The total horizontal and vertical correction should not exceed the maximum allowed correction ([DFL_H_CORR_TOL], [DFL_V_CORR_TOL]) resulting in a successful locking. This may take some time before it reach desired phase.

b. When the period error is greater than the maximum error correction, which means that the input-output timing difference is large enough. This cannot be corrected using the current tolerance limitation (otherwise it will output an unusable timing). The display synchronizer will ignore the measured phase error, and add (or drop) the maximum allowed correction per frame ([DFL_MAX_ERR_CORR]) which will get translated to a multiple [DFL_H_CORR_TOL] and one [DFL_V_CORR_TOL] every frame during the predefined locations (internally calculated).

The system remains in this state until the input change and the error become small enough for the synchronizer to switch back to stat A. It is also possible that the system will stay in this state because the input-output difference is large and not changing which mean that the display will never catch up with the input. During this mode the display is NOT LOCKED to the input and the display is in a FREE RUN mode. The new DTG total #pixel/frame = Host DTG total # pixel/frame +/- DFL_MAX_ERR_CORR.



State	Explanation of the input state	System behavior
A	Complete loss of source (like unpaged)	Free Run
B	Input jitters within limits (like a VCR PLAY mode)	Frame Lock
C	Input jitters exceeded limits (like a VCR Trick mode)	Free Run

Figure 81. DFL Functional Description

25.2 DISPLAY TIMING PROGRAMMING

Display timing signals provide timing information so the Display Output Interface can be connected to an external display device via a TTL or LVDS interface. Based on values programmed in registers, the Display Timing Generator produces the horizontal sync (DHS), vertical sync (DVS), and data enable (DEN) control signals. The figure below provides the registers that define the output display timing.

Horizontal values are in single-pixel increments. When the display is in double wide mode, horizontal settings should use even numbers. Vertical values are programmed in line increments relative to the leading edge of the vertical sync signal.

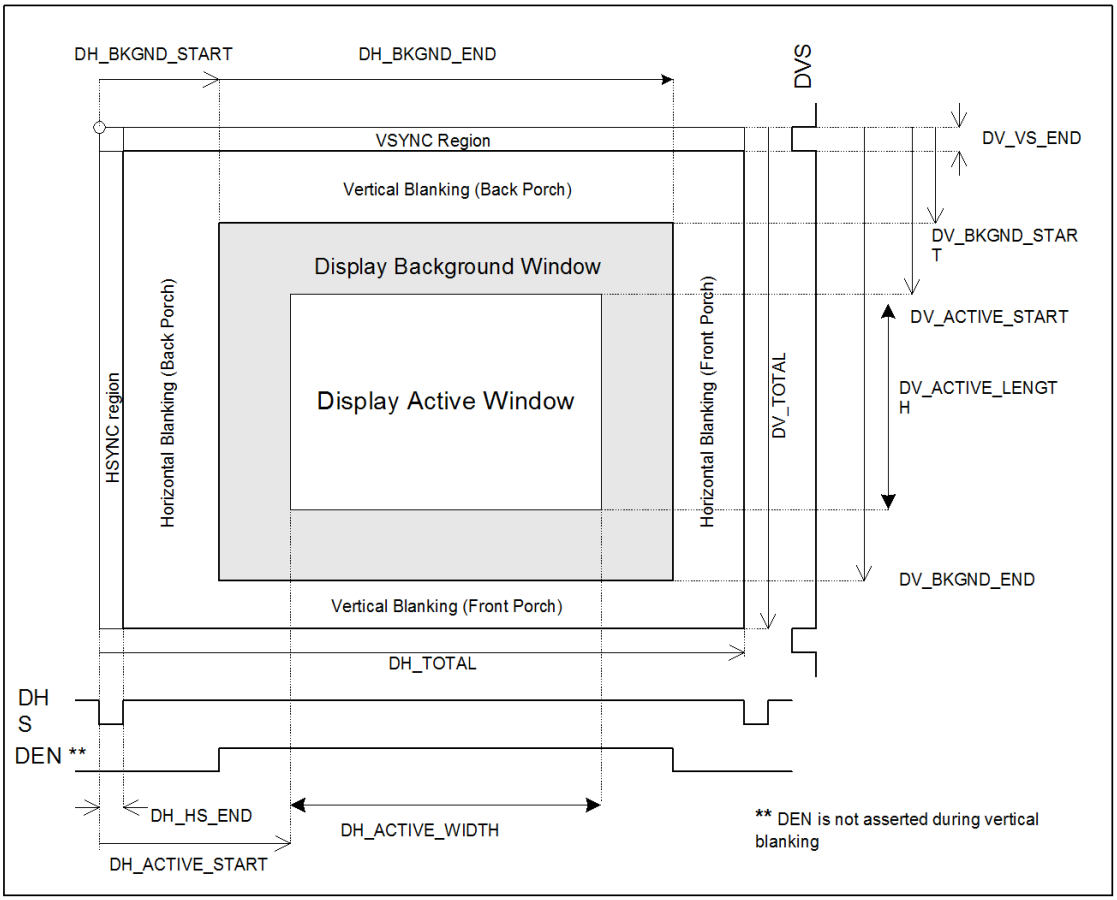


Figure 82. Display Windows and Timing

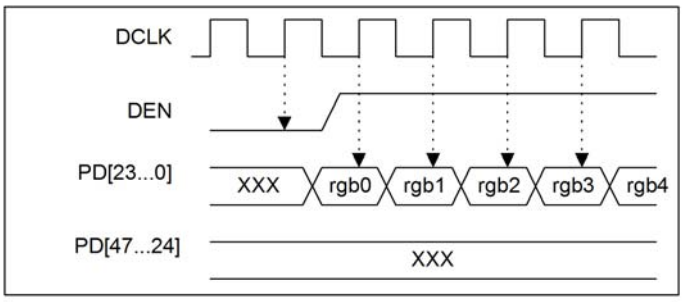


Figure 83. Single Pixel-Wide Display Data

To improve routing on the system PCB, the FLI106xx supports swapping of display pixel data.

25.3 LVDS TRANSMITTER

Two LVDS channels (A and B) are available on the output of the FLI106xx to transmit data and timing information to the display device.

Note: For single-wide LVDS usage, the LVDS even channel must be used.

FLI106xx directly drives the standard LVDS interface panels, supporting all standard data formats—single and dual bus, 18-bit, 24-bit, or 30-bit data output. The 24-bit data may be mapped as either standard receiver formats. The following diagrams illustrate the RGB, HSync, VSync, and Data Enable signal mapping in a single bus output configuration.

For dual bus output, the only difference is that the even bus contains only the "even" pixels and the odd bus contains only the "odd" pixels with the data clock at DCLK/2.

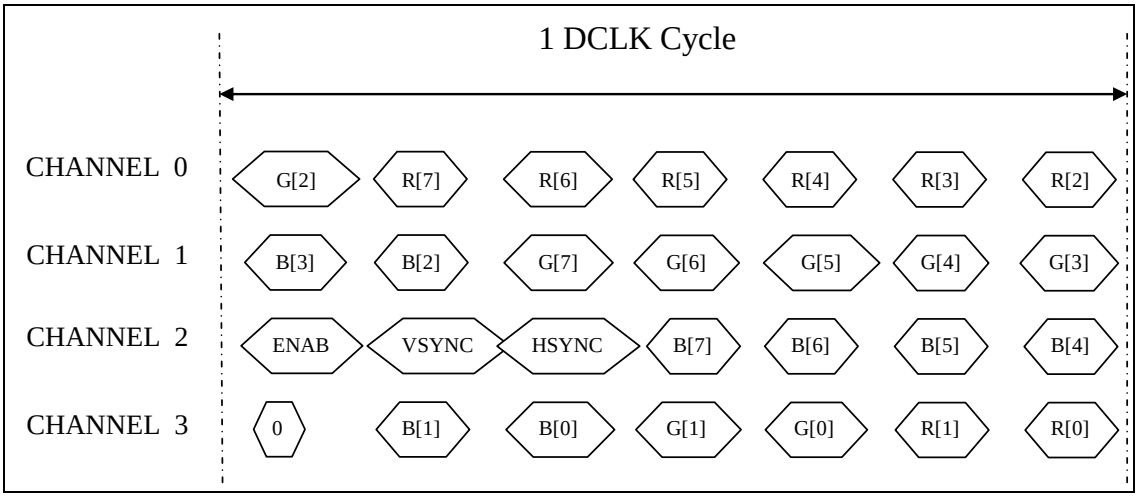


Figure 84. Data Mapping for LVDS Output in 8-bit Config. with EIGHT_BIT_MODE_SEL=0

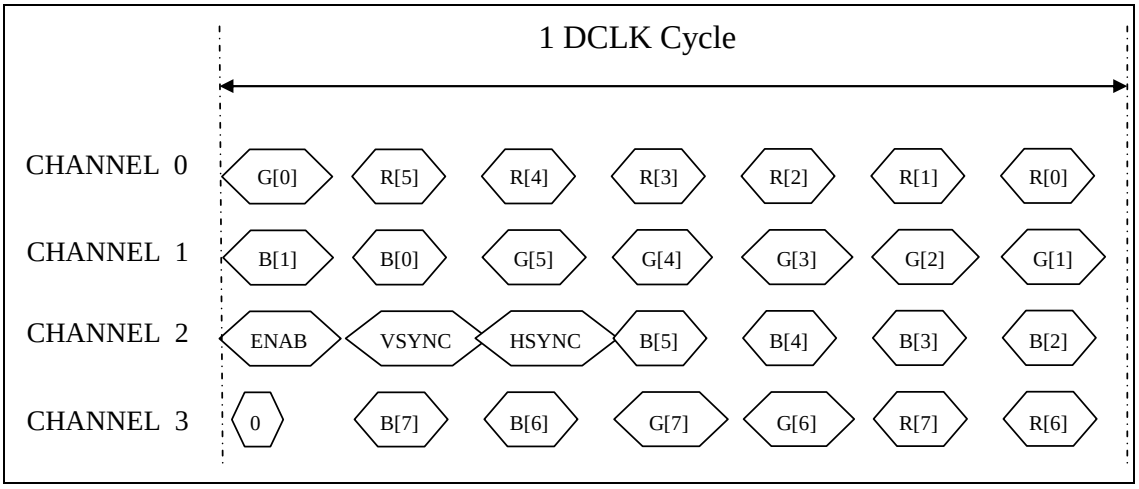


Figure 85. Data Mapping for LVDS Output in 8-bit Config. with EIGHT_BIT_MODE_SEL=1

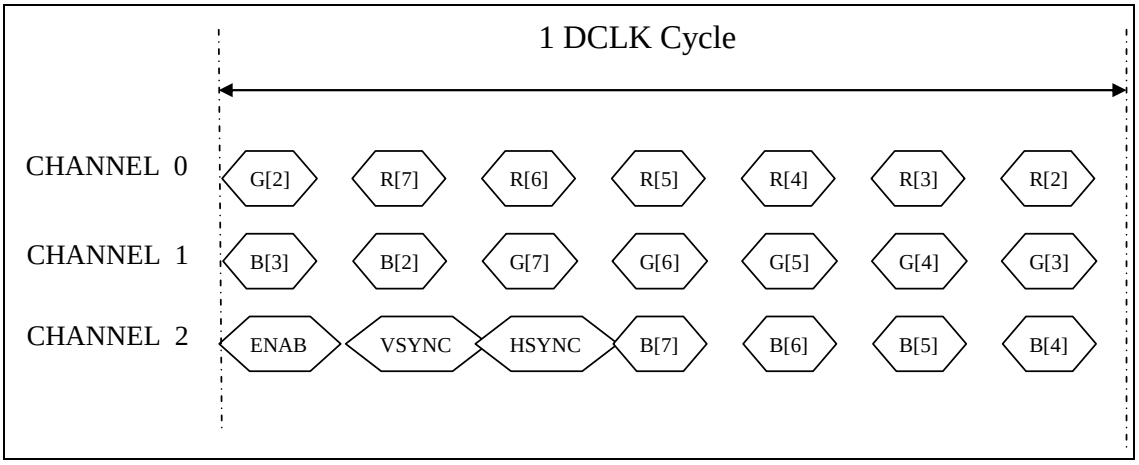


Figure 86. Data Mapping for LVDS Output in 6-bit Config. with EIGHT_BIT_MODE_SEL=0,LVDS_CH3_EN=0

For system board layout flexibility there are several data swapping options. These data swapping options are independent of the data bit formatting options. One example is shown below:

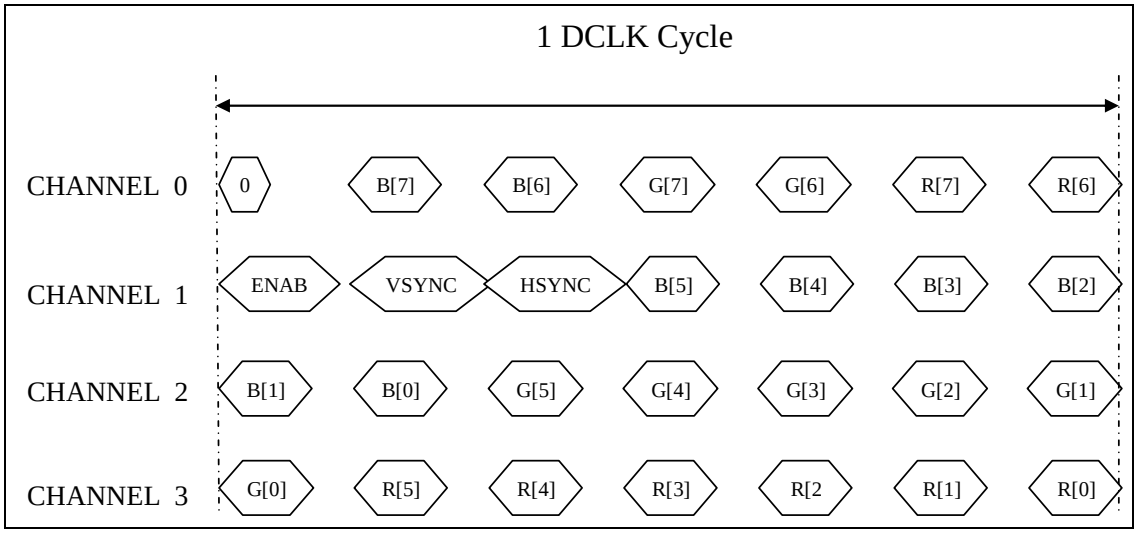


Figure 87. Data Mapping for LVDS Output in 8-bit Config. with EIGHT_BIT_MODE_SEL=1,CHANNEL_SWAP=1

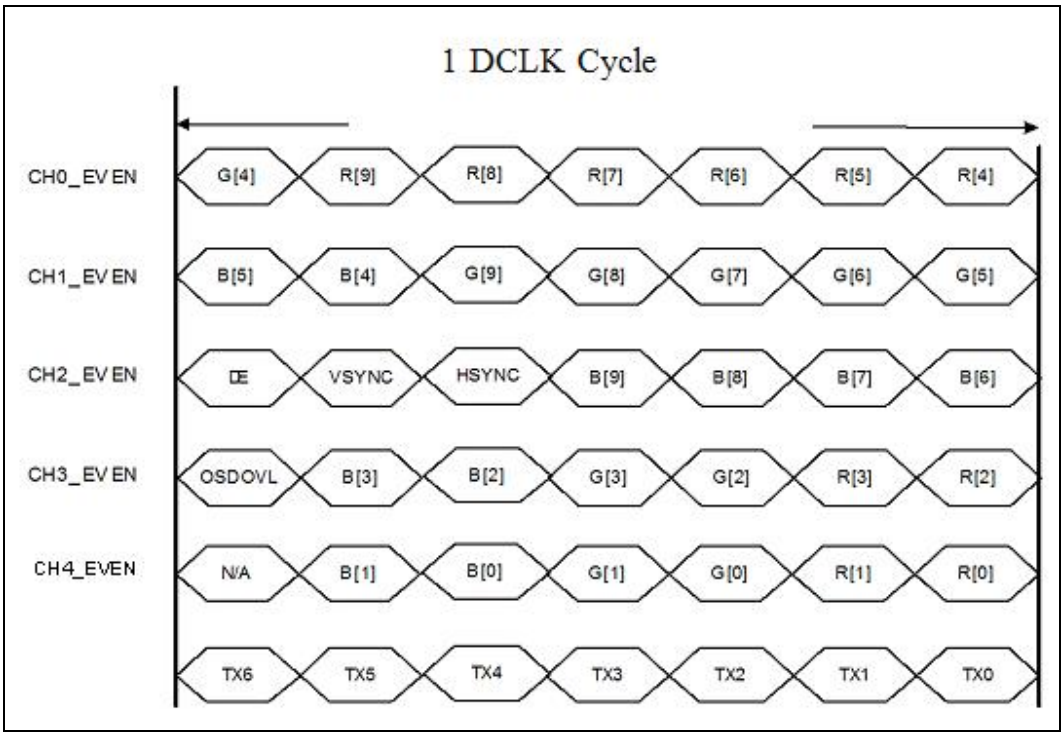


Figure 88. 30-bit LVDS Output Stream (with 0x8728[5] = '0')

25.3.1 CUSTOM LVDS MAPPING

It is possible to fully customize LVDS mapping through a bit-by-bit mapping option for those panels that do not support any of the above mappings.

25.4 PANEL POWER SEQUENCING (PPWR, PBIAS)

The FLI106xx has two dedicated outputs—PPWR and PBIAS—to control LCD power sequencing once data and control signals are stable. The timing of these signals is fully programmable.

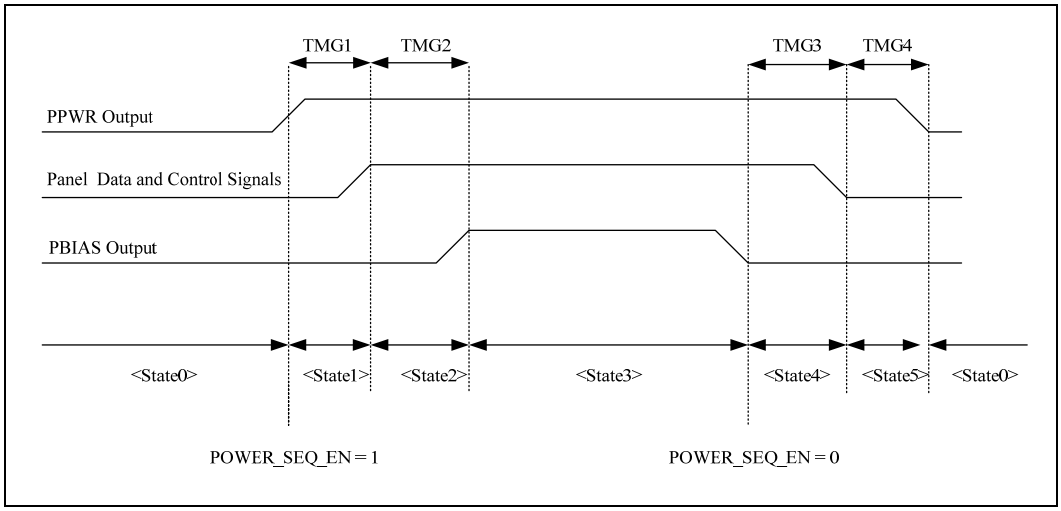


Figure 89. Panel Power Sequencing

25.5 DISPLAY COMPOSITE SYNC OUTPUT

FLI106xx supports true composite sync output. This can be made available through the LVDS link for LVDS applications or on the DHS pin for TTL applications.

Supported modes include OR-type, XOR-type, Serration-rise with/without half line serration pulses, and Serration-fall with/without half line serration pulses.

The target application is for interlaced output format to the input of a DAC.

25.6 OUTPUT DITHERING

The Gamma LUT outputs a 12-bit value for each color channel. This value is dithered down to either 10 bits for 30-bit per pixel displays, 8 bits for 24-bit per pixel displays, or 6 bits for 18-bit per pixel displays.

The human eye tends to average neighboring pixels and, therefore, a smooth image free of contours is perceived. Dithering works by spreading the quantization error over neighboring pixels both spatially and temporally.

25.7 VIDEO WINDOW

The video window is used to define an area on the main channel in which different processes can be performed inside and outside the window. The signal generated by this block is propagated through and used by the ACC II, ACM-3D, and 3x3 color matrix blocks. The ACC II and ACM-3D blocks can each be individually programmed for inside the window, outside the window, or the entire window.

25.8 ENERGY SPECTRUM MANAGEMENT® (ESM®)

High spikes in the Electromagnetic Interference (EMI) power spectrum can cause LCD monitor products to violate emissions standards. The FLI106xx has features that can be used to reduce EMI. These include:

- LVDS transmitter
- Dual-edge clocking DDR memory interface
- Single-crystal clock source
- Variable-frequency display clock control

These features eliminate the costs associated with EMI.

26 AUDIO OUTPUT

The FLI106xx chip has five baseband audio DACs, three stereo I2S audio outputs, and one S/PDIF compliant output. Audio data in DRAM is transferred to AUDO via DMA bus. There are two DMA channels allocated for AUDO. The first DMA channel is the main channel through which multi-channel audio samples can be sent to AUDO. The second DMA channel is the auxiliary channel through which a stereo pair of audio samples can be sent to AUDO. Thus AUDO supports one main multi-channel audio output and one stereo audio output with different sampling rate. Audio input can be bypassed directly to audio output through I2S and SPDIF transmitters. I2S output Data can be sent out either in right/left justified format. It is also possible to down mix the 5.1 channel audio stream and send it as stereo output. Tristate control for I2S and SPDIF outputs is provided via register programming.

- DAC, I2S and SPDIF always generate 48 kHz audio output. Audio input with a sampling rate other than 48 kHz is resampled in FLI106xx before sending out through AUDO. The only exception is analog and digital bypass modes.
- Some of the example usage models of the five DACS are as follows:
    a. 2 DACs for Main L/R, 2 DACs for Center and Subwoofer, and 1 for analog VCR out
    b. 2 DACs for Front L/R, 2 for Rear L/R, and 1 for Center (or Subwoofer)
    c. 2 DACS for Main L/R and 1 for Center (or Subwoofer), and 2 for LineOut/VCR-Out
- All three I2S transmitters are used to replace the five internal DACs as a customer option to use higher quality external DACs instead.
- One of the three I2S transmitters can be enabled simultaneously with the five DACs. This I2S transmitter is used as an additional output in parallel with the five DAC outputs. Alternatively, the same I2S transmitter is used for VCR audio output. In this case, this I2S transmitter is running independently with different clock source from the five DAC clock source.
- SPDIF transmitter is used to transmit either compressed 5.1 stream or stereo PCM.

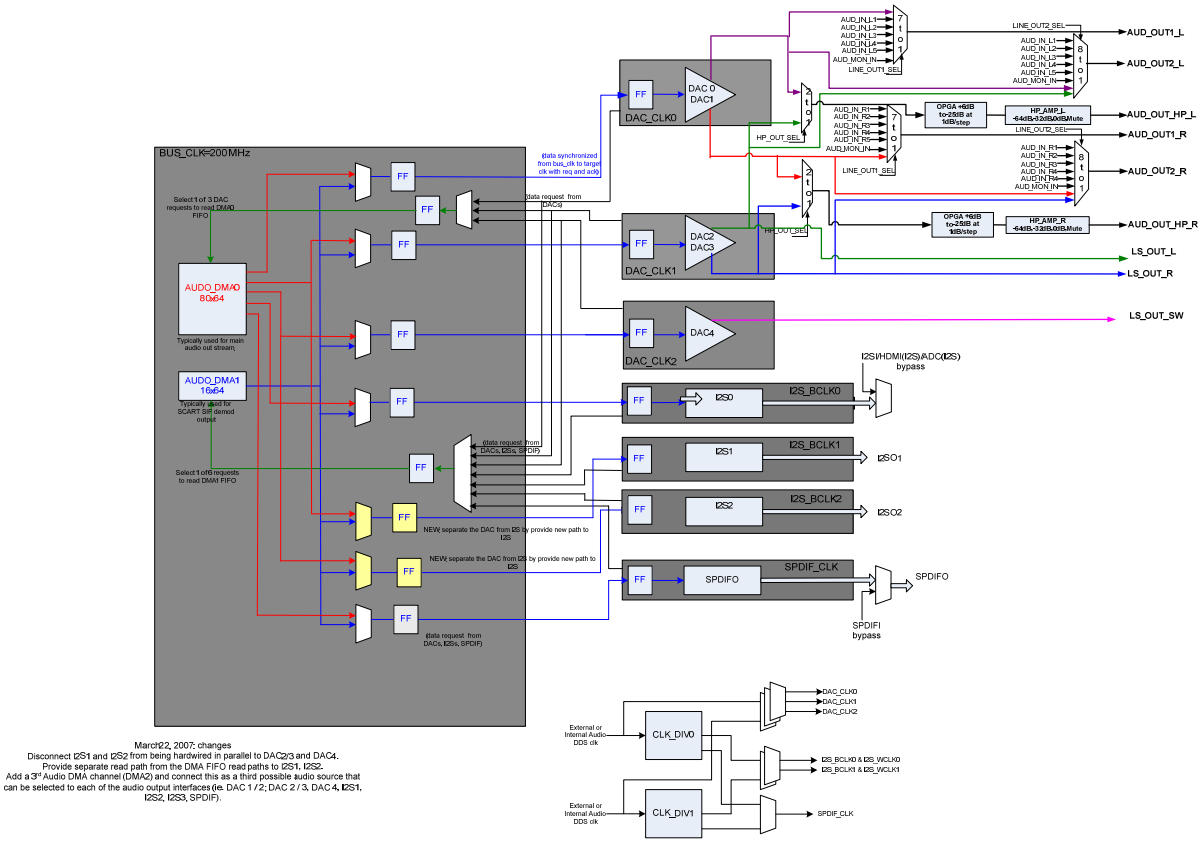


Figure 90. Audio Out Block Diagram

26.1 AV SYNCHRONIZATION

The FLI106xx chip can interact with software to synchronize audio and video outputs. For instance, the chip can be configured to ensure that the sound program never leads the video program by more than 5 milliseconds and never lags the video program by more than 15 milliseconds. Sometimes this requires additional delay insertion in the audio decode path and FLI106xx has the capability to insert additional and separate audio delay in the main and secondary audio paths.

27 VCR OUT

The VCR Out (VCRO) block provides a secondary output for an external VCR recorder that allows "watch and record" of the main channel.

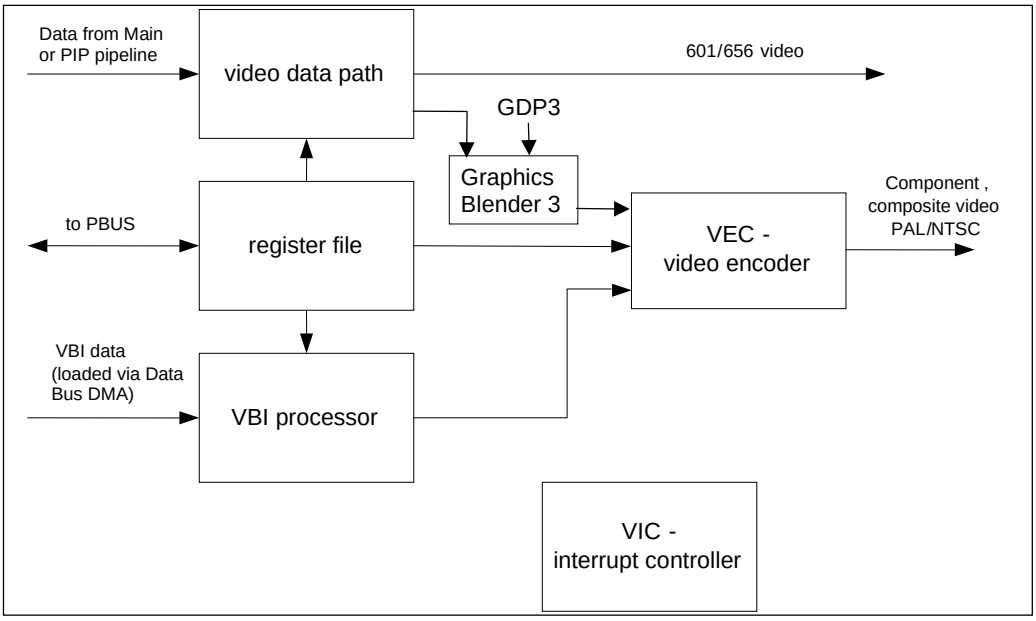


Figure 91. VCRO Block Diagram

27.1 INPUT SOURCE CAPTURE

27.1.1 INPUT SELECTION

- Uses Main channel OR PIP channel as input
- DTV inputs (from transport stream in) supported on VCR output (when DTV input is selected as source to FLI106xx main/PIP channel.)
- Standard definition analog video inputs to be routed around FLI106xx and muxed to VCR out using analog video switch in the on-chip Analog Front End
- PAL source to NTSC VCR out not supported
- NTSC source to PAL VCR out not supported

27.1.2 INPUT CROPPING

- Can crop input image independently from main/PIP channel, using separate cropping window registers.
- Cropping is programmed in the same context as the IMP (input main channel) for easier (symmetrical) programming with the input main channel set-up/configuration.

27.2 IMAGE PROCESSING (COLOR, SCALING, FRC, ETC.)

- Input Color Space Conversion
- Convert 4:4:4 RGB (0-255) to YCbCr (16-Y:235/CbCr:240)
- Convert 4:4:4 YPbPr (16-Y:235/PbPr:240) to 16-Y:235/CbCr:240
- Convert 4:2:2 YPbPr (16-Y:235/PbPr:240) to 16-Y:235/CbCr:240

Note: 4:2:2 or 4:4:4 YCbCr input requires no color conversion.

In Figure 92 below, Main or PIP video, coming from memory, can be chosen to be processed by VCRO video block. Also coming from memory can be MPEG subtitle, MHEG-5 subtitle, and teletext data, which goes to GDP3. GDP3 output is blended with Video Processing before it goes to Video DAC's or TTL interface. Note that GDP3 is only needed in DVB case. In ATSC case, the Video Processing output goes directly to VCR_out without any graphics blending with GDP3. GDP3 can change color spaces and also downsample 4:4:4 to 4:2:2 format.

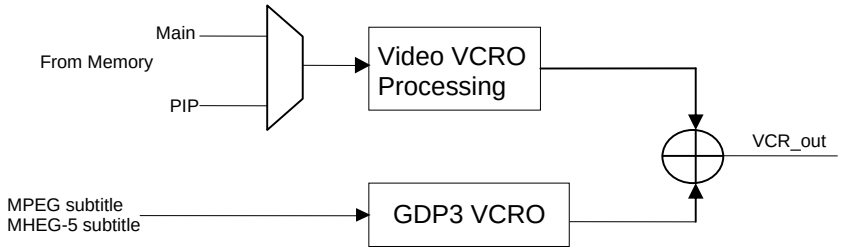


Figure 92. VCRO Processing Block Diagram

27.2.1 FRAME RATE CONVERSION

- VCR output scales the main or PIP channel source video "on the fly" to the NTSC or PAL display output resolution/timing.
- Frame rate conversion and writing VCR output stream to/from frame buffer is **not** supported.

27.2.2 ASPECT RATIO CONVERSION

- No Letterbox display support for 1080i DTV input when driven to VCR output
- Can crop a 4:3 sub-region from 16:9 1080i active region
- Can perform horizontal panoramic scale to "squeeze" the sides of a 16:9 image to fit onto 4:3 VCR output
- Bandwidth permitting (i.e., for the 720p DTV input case), could vertically compress the MDTP vertical active region time to better fit a 16:9 DTV source image onto NTSC 4:3 output display.

  Note: This increases memory bandwidth by approximately 20%, and since 720p sources don't require MADi deinterlacing, then their bandwidth consumption is much lower and, therefore, this is achievable.

27.2.3 VERTICAL SCALING

- 4 tap vertical scaler
- 1080p inputs

27.2.4 HORIZONTAL SCALING

- 6 tap horizontal scaler
- 19 sets of scaling coefficients

27.2.5 DYNAMIC CHANGES TO INPUT FORMAT

During dynamic changes in the input source format (i.e., Movie playing at 720p resolution, commercial cuts in at 480i, movie then continues at 720p resolution) the VCR output scaling and display timing will have to be reprogrammed using pending/active registers to change correctly on field-by-field basis to match the source resolution. Even with synchronized register updates, when the source resolution/timing the "start of image" reference point for display

synchronization may change significantly resulting in some "hard sync" to display timing – therefore, on the VCR output dynamic changes to input format will not be "seamless".

27.3 COMPOSITE/SVIDEO OUTPUT

27.3.1 VBI DATA INSERTION

The following VBI insertion is supported.

- NTSC or 525 line/60Hz VBI support
  - CGMS (line 20) – US and Japan
  - CC (line 21) – US
  - EDTV-II (line 22) - Japan
- PAL VBI support
  - Teletext (line 6-22) - Europe
  - VPS (line 16) – Europe
  - WSS (line 23) - Europe

27.3.2 656 OUTPUT STREAM FOR EXTERNAL VIDEO ENCODER

When the analog video output DACs are being used to drive component or RGB video to CRT display, they need to be able to provide a 656 output stream via the shared LVTTL/LVDS output port for external video encoder.

27.3.3 VIDEO ENCODER

- Macrovision support
- Video encoder will be running at 60 Hz field rate when input source is 60 Hz frame or field rate.

27.3.4 COMPOSITE/S-VIDEO ANALOG OUTPUT

Supports both S-Video and composite outputs (one at a time).

27.3.5 ANALOG MUX FOR BYPASS OF STANDARD DEF ANALOG INPUTS

Analog standard definition CVBS or S-Video (i.e., VCR or tuner source) to do analog bypass of FLI106xx device, as follows:

- AFE mux selects copy of the CVBS or S-Video to be output on AVOUT pin
- If input source is DTV and main channel display is flat panel, then drive main display with LVDS or LVTTL, and drive VCROUT data path through FLI106xx on-chip DAC. Route DAC output on system PCB back to AFE where AFE can select it to be output on AVOUT pin. AVOUT pin drives VCR output connector on back of TV set.
- If input source DTV and main channel display is CRT, then drive main display with 3 channel DAC output from FLI106xx. Route FLI106xx VCR data pat output to LVDS/TTL output port in TTL mode, and drive external video encoder with ITU656 stream. The output of the ITU656 encoder then drives one AFE analog input where it can be selected by analog mux to drive the AVOUT pin. AVOUT pin drives VCR output connector on back of TV set.

28 PERIPHERALS

28.1 UART

The FLI106xx chip contains three UART ports compatible with industry standard 16550. One of the UART is shared with 2-Wire Master. The UART transfers data between the internal data bus and external serial devices, such as keyboard, modem, etc. These ports can also be used to interface with other processors (e.g. FLI8532), upgrade and debug software, or connect to an IrDA port via an external transceiver. The UART supports either hardware or software flow control.

Features:

- 32 byte FIFOs for the Receiver and Transmitter
- Full range of Receiver and Transmitter trigger levels
- Programmable Baud Generator
- Fully programmable character formatting:
  - 5-, 6-, 7-, or 8-bit characters (Both Rx and Tx)
  - Even-, Odd-, Stick Parity- or No-Parity formats
  - 1, 1.5, or 2 stop bit (Both Rx and Tx)
- Enhanced error detection facilities:
  - Character error revealed
  - Block error revealed
  - False start-bit detection
- Line Break generation and detection (Tx only)
- Automatic hardware flow control
- Modem control functions (CTS, RTS)
- Programmable auto-RTS and auto-CTS
- Complete status reporting capabilities
- Internal diagnostic capabilities: Loop-back controls for communications link fault isolation
- External diagnostic capabilities: Echo facility

28.2 INTERRUPT CONTROLLER

The FLI106xx supports interrupt routing from interrupt sources to the MIPS 4KE processor. There are four processors in the chip: MPE0, MPE1, MPE2, and MIPS. The interrupts to each of them are described in the following sections.

The MIPS32®4KE™ Processor core receives exceptions from a number of sources, including I/O interrupts. The interrupts occur when a device outside the MIPS CPU wants attention. There are 2 categories of exception: interrupts and traps.

MIPS RISC architecture supports the concept of co-processors. Co-processor c0 is for exception handling and is always present. C0 has numerous registers, including:

- **Status Register**: Records previous interrupt settings
- **Cause Register**: How hardware makes information available to software about pending interrupts
- **Exception Program Counter (EPC)**: Loaded with the address of the instruction that was executing when the exception occurred

The EBB (Ec Bus Bridge) bridges the MIPS CPU to various processing cores. One side of EBB is connected with the MIPS core, and the other side is connected to Memory, data bus, and HIF (host interface).

28.2.1 MIPS INTERRUPTS RESOURCES

The MIPS32®4KE Processor core has a set of eight independent interrupt bits in the Cause register. This supports two software interrupts, six hardware interrupts, and a special-purpose timer interrupt.

28.2.2 MIPS INTERRUPT MODE

Interrupts Compatibility Mode: Software has one interrupt vector for all hardware interrupts and needs to prioritize them.

Vectored Interrupt (VI) Mode: Adds the ability to prioritize and vector interrupts to a handler dedicated to that interrupt and to assign a GPR Shadow set for use during interrupt processing.

28.2.3 MODULE LEVEL IMPLEMENTATION

All modules output a single active high interrupt signal and the registers operate on bus_clk only. Each module is responsible for any clock translation needed. When the software gets an interrupt, it:

- Reads the interrupt_ANDed_src register in MIPS
- Reads the interrupt_ANDed_src register in the appropriate module
- Clears the interrupt by writing '1' in the appropriate location in the interrupt_src register in that module

Each module implements the following three registers:

- Interrupts enable register: RW type register, '1' – interrupts is enabled '0' – interrupts is disabled.
- Interrupt ANDed source register. CRO type register, when overwritten with a '1' the read on bit will be cleared to '0'
- Interrupt ANDed source register: RO type register. It is bit-wise AND of the INTR_SRC and INTR_ENABLE. This is useful for the software to read only unmasked interrupt sources.

28.2.4 EBB IMPLEMENTATION

In the FLI106xx software implementation, a single vector is used by software to provide control for all cores.

For EBB register setting:

- Interrupt enable register: RW-type register,'1' – interrupts is enabled '0'– interrupts is disabled.
- Interrupt source register. RO-type register, it holds the status.
- Interrupt ANDed source register: RO-type register, it returns the value of interrupt source register ANDed with interrupt Enable Register.

In interrupt source and interrupt ANDed source register, each bit is allotted for a module.

28.2.5 INTERRUPT ROUTING TO MIPS CORE

The FLI106xx supports two levels of interrupt masking: at the module level and at the global (MIPS) level. Module level interrupt masking is done for both CRO (Clearable Read-

Only) and RO (Read Only) type registers. Module level interrupts are masked by a programmable register inside the module, ORed, and passed to MIPS. At the global level MIPS considers all interrupt lines as RO type interrupt causes. The interrupt lines are masked (by a programmable register inside MIPS), ORed and passed to MIPS.

Caution: All writes from MIPS are posted writes, i.e., they are committed after some time but the software issues an acknowledgement immediately. For example, when software writes the interrupt source to clear it after serving, it may not de-assert the interrupt line immediately. If the software is sensitive about it, it can issue a SYNC instruction after writing a register or perform a read to the same register after writing the register.

28.3 GPIO

Almost all digital and some analog pins on the FLI106xx are shared as GPIO, GPI, or GPO pins. This provides maximum flexibility for users to configure the device for their specific system application.

GPIO pins default to their normal mission mode function but can be enabled for GPIO operation.

GPIO pins, when programmed as inputs, are programmable to generate optional interrupt to CPU on a pin-by-pin basis or, can optionally be polled to detect external system events. GPIO, when programmed as outputs, can be set or cleared with separate set and clear programming registers to avoid the need for read-modify-write operations when setting individual bits in a 32-bit register.

The mapping of GPIO registers to device pins is as follows:

GPIO0	UART1_CTS	GPIO32	AUDIN_SPDIF_IN
GPIO1	UART1_RXD	GPIO33	AUD_MCLK1
GPIO2	UART1_TXD	GPIO34	AUDO_I2SA_BCLK
GPIO3	UART1_RTS	GPIO35	AUDO_I2SA_WCLK
GPIO4	PWM0	GPIO36	AUDO_I2SA_DAT0
GPIO5	PWM1	GPIO37	AUDO_I2SB_BCLK
GPIO6	PWM2	GPIO38	AUDO_I2SB_WCLK
GPIO7	PWM3	GPIO39	AUDO_I2SB_DAT1
GPIO8	DFSYNC_IN, DFSYNC_OUT	GPIO40	AUDO_I2SB_DAT2
GPIO9	HDMI_A_HPD	GPIO41	AUDO_SPDIF_OUT
GPIO10	HDMI_CEC	GPIO42	OOB_CTX
GPIO11	CLKOUT	GPIO43	OOB_CRX
GPIO12	LVTX_ODD_CH5N_DISPCLK	GPIO44	OOB_DRX
GPIO13	LVTX_ODD_CH5P_DISPDE	GPIO45	POD_CD_2
GPIO14	LVTX_EVN_CH5N_DISPVS	GPIO46	POD_CD_1
GPIO15	LVTX_EVN_CH5P_DISPHS	GPIO47	POD_CE_2
GPIO16	POD_DETECT_N	GPIO48	POD_WAIT_N
GPIO17	POD_DIR_N	GPIO49	POD_READY_IRQ_N
GPIO18	POD_CE_1	GPIO50	POD_RESET
GPIO19	POD_A4_CTX	GPIO51	TWOWIRE_M1_SDA/UART2_TXD
GPIO20	POD_A5_ITX	GPIO52	TWOWIRE_M1_SCL/UART2_RXD
GPIO21	POD_A6_ETX	GPIO53	TWOWIRE_S0_SDA
GPIO22	POD_A7_QTX	GPIO54	TWOWIRE_S0_SCL
GPIO23	POD_A8_CRX	GPIO55	TWOWIRE_S1_SDA
GPIO24	POD_A14_MCLKO	GPIO56	TWOWIRE_S1_SCL

GPIO25	POD_VS2_MCLKO	GPIO57	TWOWIRE_S2_SDA
GPIO26	POD_BVD2_MOVAL	GPIO58	TWOWIRE_S2_SCL
GPIO27	POD_BVD1_MOSTRT	GPIO59	POD_A9_DRX
GPIO28	AUD_MCLK0	GPIO60	HOST_DEV_CS2_N
GPIO29	AUDIN_I2S_BCLK	GPIO61	HOST_DEV_CS1_N
GPIO30	AUDIN_I2S_WCLK	GPIO62	HOST_A24
GPIO31	AUDIN_I2S_DAT	GPIO63	AUDO_MUTE

GPIO64	HOST_READY	GPIO96	POD_D8_MDO0	GPIO128	VXO_CLK
GPIO65	HOST_DEV_CS0_N	GPIO97	POD_D9_MDO1	GPIO129	VXO_DE
GPIO66	HOST_BOOT_CS_N	GPIO98	POD_D10_MDO2	GPIO130	VXO_VS
GPIO67	HDMI_B_HPD	GPIO99	POD_D11_MDO3	GPIO131	VXO_HS
GPIO68	VXI_CLK	GPIO100	POD_D12_MDO4	GPIO132	VXO_D0
GPIO69	VXI_DE	GPIO101	POD_D13_MDO5	GPIO133	VXO_D1
GPIO70	VXI_VS	GPIO102	POD_D14_MDO6	GPIO134	VXO_D2
GPIO71	VXI_HS	GPIO103	POD_D15_MDO7	GPIO135	VXO_D3
GPIO72	VXI_D0	GPIO104	LVTX_ODD_CH0N_DISP23	GPIO136	VXO_D4
GPIO73	VXI_D1	GPIO105	LVTX_ODD_CH0P_DISP22	GPIO137	VXO_D5
GPIO74	VXI_D2	GPIO106	LVTX_ODD_CH1N_DISP21	GPIO138	VXO_D6
GPIO75	VXI_D3	GPIO107	LVTX_ODD_CH1P_DISP20	GPIO139	VXO_D7
GPIO76	VXI_D4	GPIO108	LVTX_ODD_CH2N_DISP19	GPIO140	VXO_D8
GPIO77	VXI_D5	GPIO109	LVTX_ODD_CH2P_DISP18	GPIO141	VXO_D9
GPIO78	VXI_D6	GPIO110	LVTX_ODD_CLKN_DISP17	GPIO142	VXO_D10
GPIO79	VXI_D7	GPIO111	LVTX_ODD_CLKP_DISP16	GPIO143	VXO_D11
GPIO80	VXI_D8	GPIO112	LVTX_ODD_CH3N_DISP15	GPIO144	VXO_D12
GPIO81	VXI_D9	GPIO113	LVTX_ODD_CH3P_DISP14	GPIO145	VXO_D13
GPIO82	VXI_D10	GPIO114	LVTX_ODD_CH4N_DISP3	GPIO146	VXO_D14
GPIO83	VXI_D11	GPIO115	LVTX_ODD_CH4P_DISP2	GPIO147	VXO_D15
GPIO84	VXI_D12	GPIO116	LVTX_EVN_CH0N_DISP13	GPIO148	HOST_D8
GPIO85	VXI_D13	GPIO117	LVTX_EVN_CH0P_DISP12	GPIO149	HOST_D9
GPIO86	VXI_D14	GPIO118	LVTX_EVN_CH1N_DISP11	GPIO150	HOST_D10
GPIO87	VXI_D15	GPIO119	LVTX_EVN_CH1P_DISP10	GPIO151	HOST_D11
GPIIO88	VXI_D16	GPIO120	LVTX_EVN_CH2N_DISP9	GPIO152	HOST_D12
GPIO89	VXI_D17	GPIO121	LVTX_EVN_CH2P_DISP8	GPIO153	HOST_D13
GPIO90	VXI_D18	GPIO122	LVTX_EVN_CLKN_DISP7	GPIO154	HOST_D14
GPIO91	VXI_D19	GPIO123	LVTX_EVN_CLKP_DISP6	GPIO155	HOST_D15
GPIO92	VXI_D20	GPIO124	LVTX_EVN_CH3N_DISP5	GPIO156	HOST_A20
GPIO93	VXI_D21	GPIO125	LVTX_EVN_CH3P_DISP4	GPIO157	HOST_A21
GPIO94	VXI_D22	GPIO126	LVTX_EVN_CH4N_DISP1	GPIO158	HOST_A22
GPIO95	VXI_D23	GPIO127	LVTX_EVN_CH4P_DISP0	GPIO159	HOST_A23

The GPI Pin Mapping Summary is as follows:

GPI registers can be programmed on a pin-by-pin basis to interrupt the MIPS CPU.

GPI 12:0

GPI0	UART0_RXD
GPI1	UART0_TXD
GPI2	CDI0_VALID
GPI3	CDI0_CLK
GPI4	CDI0_SYNC
GPI5	AHS_ACS
GPI6	AVS
GPI7	HOST_ACK
GPI8	IRDATA
GPI9	CDI0_D0
GPI10	SCART_FB
GPI11	USB_FLAG
GPI12	CDI0_ERROR

The mapping of GPO Registers is as follows:

GPO pins also support the set and clear register programming functions as GPIO pins for ease of single bit manipulation.

GPO 29:0

GPO0	LBADC_IN1
GPO1	LBADC_IN2
GPO2	LBADC_IN3
GPO3	LBADC_IN4
GPO4	LBADC_IN5
GPO5	LBADC_IN6
GPO6	PBIAS
GPO7	PPWR
GPO8	POD_HOST_A0
GPO9	POD_HOST_A1
GPO10	POD_HOST_A2
GPO11	POD_HOST_A3
GPO12	PODREG_HOST_A4
GPO13	POD_IOWR_HOST_A5
GPO14	POD_IORD_HOST_A6
GPO15	HOST_A7
GPO16	HOST_A8
GPO17	HOST_A9
GPO18	POD_HOST_A10
GPO19	POD_HOST_A11
GPO20	POD_HOST_A12
GPO21	POD_HOST_A13
GPO22	HOST_A14

GPO23	HOST_A15
GPO24	HOST_A16
GPO25	HOST_A17
GPO26	HOST_A18
GPO27	HOST_A19
GPO28	POD_WE_HOST_WR
GPO29	POD_OE_HOST_RD
GPO30	USB_PWREN

28.4 PWM

FLI106xx provides four pulse-width modulation (PWM) outputs. These can be used to generate analog voltage levels to control various components on the board. The PWM outputs can be driven out on the following pins:

- PWM3 A13
- PWM2 B13
- PWM1 C13
- PWM0 D13

Two of the PWMs can be used to drive the analog control voltage of an external VCXO (See MPEG Clock Recovery section in MPEG Video Decoding). The duty cycle of these PWMs are set via two 10-bit counters representing the high value and the total value. The third PWM can be used to generate backlight brightness control for LCD panels.

This block generates the PWM signal using an 8-bit counter. The counter should be able to work on the TCLK, TCLK/512 or the display HSYNC. There is an option to divide the selected clock by 1, 2, 4 or 8. The pulse width and the period are programmable. An option is provided for bypassing the programmed pulse width and period and generating a precise 1024 count period (using a 10-bit counter) with the programmed pulse and period. An option to reset the counter on the rising edge of VSYNC instead of continuous rollover is available. This helps in generating the PWM signal in sync with the display VSYNC.

28.5 2-WIRE

The on-chip 2-wire bus consists of a serial clock and bi-directional serial data line. The bus is used for connection and control of other system chips (tuner, demodulator, display processor, NTSC decoder, HDMI), front panel interface, EEPROM, and real-time clock. The FLI106xx chip contains four instances of 2-wire. Two are configured as master while the others are slaves. One of the 2-Wire masters is shared with a UART. The master interface is connected to front-end components like tuner and demodulator, while the slave interface is connected to back-end processors (like FLI8532, HDMI chip, DACs) and EEPROM.

A two-wire data transfer consists of a stream of serially transmitted bytes formatted as shown in the following figure. A transfer is initiated (START) by a high-to-low transition on data line while clock is held high. A transfer is terminated by a STOP (a low-to-high transition on data line while clock is held high) or by a START (to begin another transfer).

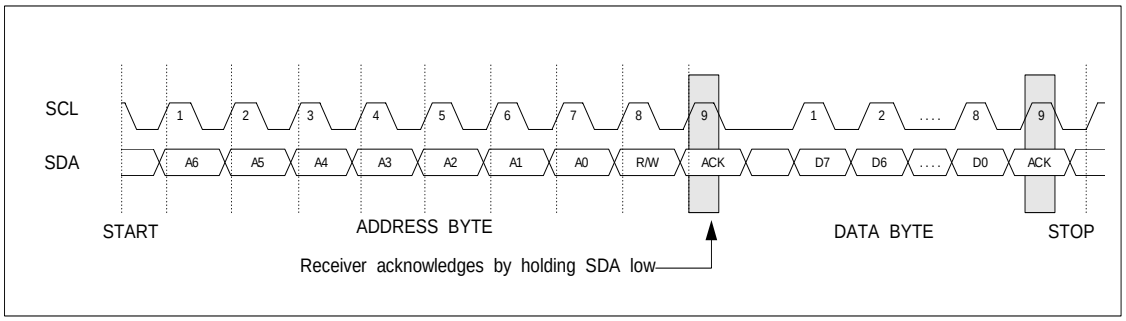


Figure 93. Two-Wire Protocol Data Transfer

Each transaction on the SDA is in integer multiples of 8 bits (i.e., bytes). The number of bytes that can be transmitted per transfer is unrestricted. Each byte is transmitted with the most significant bit (MSB) first. After the eight data bits, the master releases the MSTR_SDA line and the receiver asserts the SDA line low to acknowledge receipt of the data. The master device generates the SCL pulse during the acknowledge cycle. The addressed receiver is obliged to acknowledge each byte that has been received.

Features:

- Supports data communication that is fully controllable by software
- Enables normal mode and fast mode of up to 400 KHz
- Supports bi-directional data transfer between masters and slaves
- Provides a programmable spike filter that filters spikes on the bus data line to preserve data integrity
- Supports serial clock synchronization that allows communication between devices with different bit rates

28.6 IR RECEIVER

The IR receiver is intended to receive commands from a remote controlled device for changing channels and navigation. The receiver can be programmed to process any of the following protocols: RC5, REC80 and NEC. It generates a command interrupt at the beginning of a command and a data interrupt whenever there is data in the FIFO.

Features:

- Built-in demodulator
- 32 byte deep FIFO
- Full Interrupt Support
- Error detection and reporting
- Programmable Noise Filter

28.6.1 REC-80 (PANASONIC)

The REC-80 (Panasonic) standard varies the length of the spaces between pulses to encode IR data. In this method, when the space width is short (approximately 550us) the signal corresponds to a logical zero or a low. When the space width is long (approximately 1650us) it corresponds to a logical one or a high.

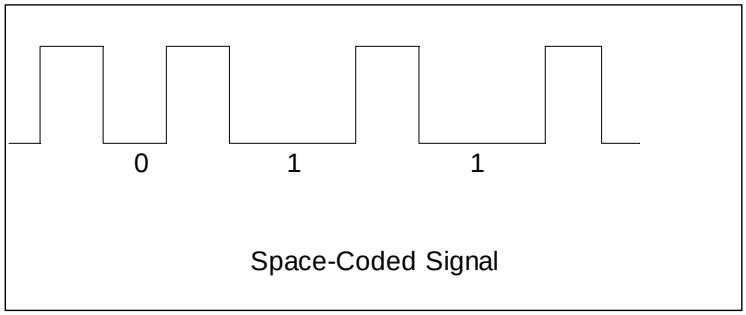


Figure 94. Space-Coded Signal

28.6.2 RC-5 (PHILIPS)

The RC-5 (Philips) standard varies the order of pulse space to code the information. In this protocol when a short space (approximately 550us) is followed by a long pulse (approximately 1100us) the signal corresponds to a logical one or a high. On the other hand, when a long space is followed by a short pulse, the signal corresponds to a logical zero or a low. This coding holds information such as the address to the machine that is using the remote and the command that the machine must follow.

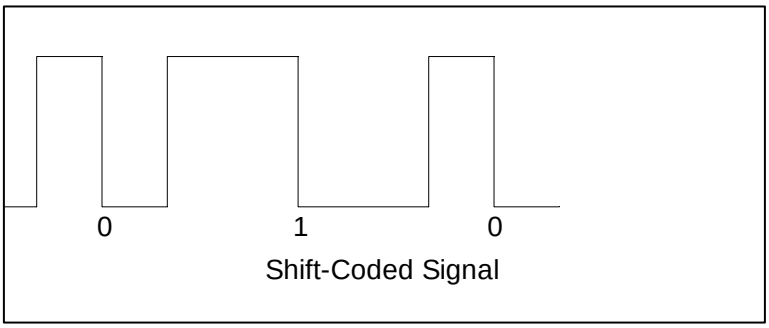


Figure 95. Shift-Coded Signal

28.6.3 NEC

The NEC standard varies the length of pulses to code the information. In this protocol when the pulse width is short (approximately 550us) it corresponds to a logical zero or a low. When the pulse width is long (approximately 2200us) it corresponds to a logical one or a high.

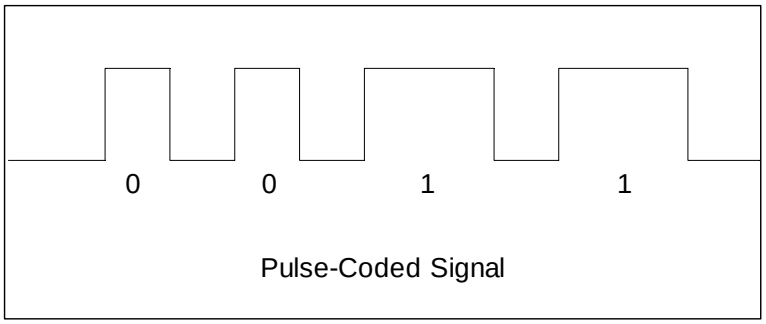


Figure 96. Pulse-Coded Signal

28.7 JTAG/EJTAG

The FLI106xx includes a shared JTAG/EJTAG interface that provides the means for software debug, boundary-scan, and board bring-up.

28.7.1 EJTAG

The EJTAG interface operates through the Test Access Port (TAP), a serial communication port used for transferring test data in and out of the MIPS processor. In addition to the standard JTAG instructions, special instructions defined in the EJTAG specification define what registers are selected and how they are used.

The MIPS processor provides an Enhanced JTAG (EJTAG) interface for software debugging of application and kernel code. In addition to standard user mode and kernel modes of operation, the 4KEc core provides a Debug mode that is entered after a debug exception (derived from a hardware breakpoint, single-step exception, etc.) is taken and continues until a debug exception return (DERET) instruction is executed. During this time, the processor executes the debug exception handler routine.

The EJTAG block allows for single-stepping of the processor, as well as instruction and data virtual address/value breakpoints. Additionally, real-time tracing of instruction program counter, data address, and data values is supported.

28.7.2 JTAG

The FLI106xx supports the IEEE 1149.1 JTAG boundary scan standard. The chip provides a tap controller and boundary scan cells to monitor/drive the chip interfaces. This feature can be used for board level testing.

The DDR pins are not covered by JTAG. However, a mechanism is provided to interrupt MIPS over JTAG, which can then be used to execute a DDR test program from the FLASH.

28.8 LOW BANDWIDTH ADC

The low bandwidth ADC can be used to convert low speed analog signals to digital. It is integrated to allow for functions such as keypad scanning, or for monitoring system temperature, or voltage sensors. The ADC has 8 bits of resolution and can perform a conversion in 13 TCLK periods (approximately 1 µsec). An analog multiplexer selects one of six analog input pins as the input to the ADC.

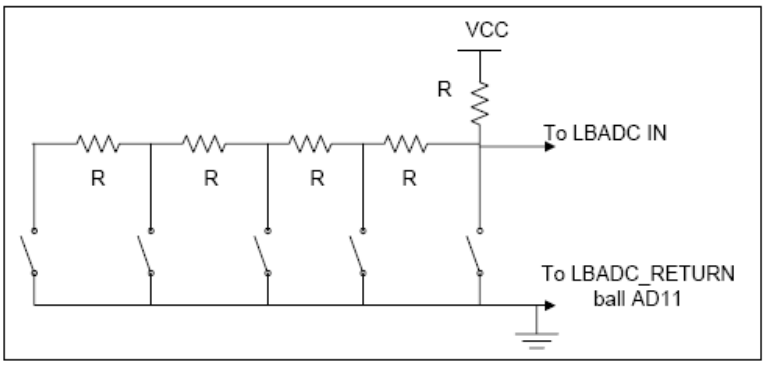


Figure 97. Typical Keypad Function

28.9 USB 2.0 HOST CONTROLLER

The FLI106xx USB 2.0 host controller contains the digital logic and analog circuitry necessary to connect to a full/low speed USB peripheral device. The USB Host Subsystem is fully compliant with the USB 2.0 specification, Enhanced Host Controller Interface (EHCI) specification (version 1.0), and Open Host Controller Interface (OHCI) specification (version 1.0a). The controller supports high-speed, 480-Mbps transfers (40 times faster than USB 1.1 full-speed mode), as well as full and low speeds through an integrated OHCI Host Controller. The USB Host subsystem integrates fully compliant USB 2.0 transceiver.

- Compliant with universal serial bus specification revision 2.0 (data rate: 1.5/12/480 Mbps)
- Compliant with open host controller interface specification for USB release 1.0a
- Compliant with enhanced host controller interface specification for USB revision 1.0
- Consists of one OHCI host controller core for full-/low-speed signaling and one EHCI host controller core for high-speed signaling
- Root hub with 1 downstream facing port, which is shared by OHCI and EHCI host controller cores
- Port can handle high-speed (480 Mbps), full-speed (12 Mbps), and low-speed (1.5 Mbps) transactions
- A built-in transceiver features programmable USB connect and disconnect

29 ELECTRICAL SPECIFICATION

The following electrical specifications apply to FLI106xx.

29.1 PRELIMINARY DC CHARACTERISTICS

Table 48. Absolute Maximum Ratings

Parameter	Symbol	Min	Typ	Max	Units
3.3 V Supply Voltages [a, b, c]	$V_{VDD\_3.3}$	-0.3	-	3.63	V
1.8 V Supply Voltages [a, b, c]	$V_{VDD\_2.5}$	-0.3	-	1.98	V
1.2 V Supply Voltages [a, b, c]	$V_{VDD\_1.8}$	-0.3	-	1.32	V
Input Voltage (5 V tolerant inputs) [a, b]	$V_{IN5Vtol}$	-0.3	-	5.5	V
Input Voltage (non 5 V tolerant inputs)	V_{IN}	-0.3	-	3.63	V
Electrostatic Discharge (HBM)	V_{ESD}	-	±2kV	-	kV
Electrostatic Discharge (MM)	V_{ESD}	-	±200V	-	V
Latch-up	I_{LA}	-	±200	-	mA
Ambient Operating Temperature	T_A	0	-	70	°C
Storage Temperature	T_{STG}	-40	-	125	°C
Operating Junction Temp.	T_J	0	70	125	°C
Thermal Resistance (Junction to Ambient) Vertical System PCB	$\theta_{JA\ (VER)}$	-	TBD	-	°C/W
Thermal Resistance (Junction to Ambient) Horizontal System PCB	$\theta_{JA\ (HOR)}$		11.6		°C/W
PSI (J-T) (Junction-to-Top-Center Thermal Characterization Parameter) Vertical System PCB	$\Psi_{JA\ (VER)}$		TBD		°C/W
PSI (J-T) (Junction-to-Top-Center Thermal Characterization Parameter) Horizontal System PCB	$\Psi_{JA\ (HOR)}$		2.8		°C/W
Thermal Resistance (Junction to Case)	θ_{JC}	-	3.5	-	°C/W
Soldering Temperature LF (30 sec.)	T_{SOL}	-	-	250	°C

a. All voltages are measured with respect to GND.

b. Absolute maximum voltage ranges are for transient voltage excursions.

c. There is no minimum requirement for the delay between 3.3V and 1.2V rails; however, 3.3V should be ahead of 1.2V otherwise it may lead to latch-up and damage the chip.

Note: Good thermal design practices should be followed to allow efficient heat removal from the system. Follow the steps in the FLI106XX System Layout Guidelines (C106XX-SLG-01) to keep the system thermally efficient.

REMINDER: Customers are required to check Tc (case top temperature) at the extremes of their system design operating conditions to ensure they have adequate heat removal and that Tj (junction temperature) is not exceeding 125C. Tj can be calculated with the PSI (J-T) parameter provided above.

Table 49. DC Characteristics

Power	Symbol	Min	Typ	Max	Units
Power Consumption @ Low Power Mode	P_{LP}	0	0	0	W
3.3 V Supply Voltages (AVDD and RVDD)	$V_{VDD\_3.3}$	3.135	3.3	3.465	V
1.8 V Supply Voltages (VDD)	$V_{VDD\_2.5}$	1.71	1.8	1.89	V
1.2 V Supply Voltages (VDD and CVDD)	$V_{VDD\_1.8}$	1.14	1.2	1.26	V
Supply Current					
1.8V DDR2 IO voltage supply	$V_{1.8\_VDD}$	TBD	246.4	TBD	mA
1.2V Core digital voltage supply	$V_{1.2\_VDD}$	TBD	1532.1	TBD	mA
1.2V Analog PLL voltage supply	$V_{1.2\_VDD}$	TBD	11.8	TBD	mA
1.2V DDR Analog PLL voltage supply	$V_{1.2\_VDD}$	TBD	8.6	TBD	mA
1.2V HDMI/DVI digital voltage supply	$V_{1.2\_VDD}$	TBD	3.8	TBD	mA
1.2V USB Core voltage supply	$V_{1.2\_VDD}$	TBD	7.3	TBD	mA
1.2V Audio digital voltage supply	$V_{1.2\_VDD}$	TBD	11.8	TBD	mA
1.2V Video DAC digital voltage supply	$V_{1.2\_VDD}$	TBD	5.9	TBD	mA
1.2V ADC digital voltage supply	$V_{1.2\_VDD}$	TBD	14.7	TBD	mA
3.3V IO voltage supply	$V_{3.3\_VDD}$	TBD	45.4	TBD	mA
3.3V DDR input Rx VDD voltage supply	$V_{3.3\_VDD}$	TBD	19.8	TBD	mA
3.3V USB digital voltage supply	$V_{3.3\_VDD}$	TBD	50.3	TBD	mA
3.3V DLL VAA0 voltage supply	$V_{3.3\_VDD}$	TBD	6.8	TBD	mA
3.3V DLL VAA1 voltage supply	$V_{3.3\_VDD}$	TBD	4.1	TBD	mA
3.3V Analog Ref clock PLL voltage supply	$V_{3.3\_VDD}$	TBD	4.5	TBD	mA
3.3V Analog DDR clock PLL voltage supply	$V_{3.3\_VDD}$	TBD	10.9	TBD	mA
3.3V LVDS Tx VDD voltage supply	$V_{3.3\_VDD}$	TBD	85.6	TBD	mA
3.3V Analog LVDS clock PLL voltage supply	$V_{3.3\_VDD}$	TBD	8.4	TBD	mA
3.3V LBADC Analog voltage supply	$V_{3.3\_VDD}$	TBD	5.3	TBD	mA
3.3V ADC Analog voltage supply	$V_{3.3\_VDD}$	TBD	181.8	TBD	mA
3.3V Video DAC Analog voltage supply	$V_{3.3\_VDD}$	TBD	128.5	TBD	mA
3.3V Audio voltage supply	$V_{3.3\_VDD}$	TBD	39.8	TBD	mA
3.3V Audio Head Phone voltage supply	$V_{3.3\_VDD}$	TBD	15.5	TBD	mA

3.3V HDMI/DVI voltage supply	$V_{3.3\_VDD}$	TBD	94.0	TBD	mA
Inputs					
High Voltage	V_{IH}	2.0	-	5.5	V
Low Voltage	V_{IL}	-0.3	-	0.8	V
High Current (V_{IN} = TBD V)	I_{IH}	-	-	±10	µA
Low Current (V_{IN} = TBD V)	I_{IL}	-	-	±10	µA
Capacitance (V_{IN} = TBD V)	C_{IN}	-	-	5	pF
Outputs					
High Voltage (I_{OH} = TBD mA)	V_{OH}	2.4	-	-	V
Low Voltage (I_{OL} = TBD mA)	V_{OL}	-	-	0.4	V
Tri-State Leakage Current	I_{OZ}	-	-	±10	µA

29.2 PRELIMINARY AC CHARACTERISTICS

Table 50. AC Timing Details

Interface	Type[1]	Active Edge	Min	Max	Unit	Signals
Audio In (reference clock: AUDIN_BCLK)	tsu	Rising Edge	0	-	nS	AUDIN_DAT AUDIN_WCLK
	th		3	-	nS	
	tclk			25	MHz	
5.1 Audio Out (reference clock: AUDOUT_5DOT1_BCLK)	tco	Falling Edge	-2	2	nS	AUDOUT_5DOT1_DATA, AUDOUT_5DOT1_WCLK
Stereo Audio Out (reference clock: AUDOUT_STEREO_BCLK)	tco	Falling Edge	-2	2	nS	AUDOUT_STEREO_DATA, AUDOUT_STEREO_WCLK
CDI0 Free Program Mode (reference clock: CDI0_CLK)	tsu	Rising Edge	.05	-	nS	CDI0_VALID CDI0_SYNC CDI0_DATA_[7:0]
	th		3.5	-	nS	
	tclk			80 (Serial)	MHz	
CDI0 CA/POD Mode (reference clock: CDI0_CLK)	tsu	Rising Edge	3	-	nS	CDI0_VALID CDI0_SYNC CDI0_DATA_[7:0]
	th		1	-	nS	
	tclk			9	MHz	
DDR (DQ) (reference clock: DDR_CK)	tsu	Rising Edge / Falling Edge	100	-	ps	DDR_DQ[31:0] DDR_DQS_[3:0] DDR_DM[3:0]
	th		175	-	ps	
DDR (Addr and Ctl) (reference clock: DDR_CK)	tsu	Rising Edge	200	-	ps	DDR_A_[13:0] DDR_CKE DDR_CAS_N DDR_RAS_N DDR_WE_N DDR_CS_N DDR_BA[1:0]
	th		275	-	ps	
Digital Video Out (reference clock: VIDEO1_CLK_OUT)	tco	Rising Edge	1	4.5	nS	VIDEO1_OUT_[15:0] VIDEO1_DE VIDEO1_VSYNC VIDEO1_HSYNC
	tco	Falling Edge	1	4.5	nS	VIDEO1_OUT_[15:0] VIDEO1_DE VIDEO1_VSYNC VIDEO1_HSYNC

- [1] tsu: Setup time with respect to reference clock
- th: Hold time with respect to reference clock
- tclk: reference clock frequency
- tco: Clock to output delay with respect to reference clock

Video 2 Output (reference clock: DISP_27_GPIO_B3)	tco	Falling Edge	-2.5	2	nS	DISP_[47..24]_GPIO_B23_0[23:0],
LVDS, Host Interface	The setup and hold time for LVDS and Host Interface is programmable.					

29.3 DDR2 Memory Interface Timings

The following timing diagrams and tables describe the DDR2 interface timings currently supported by the FLI106xx chip.

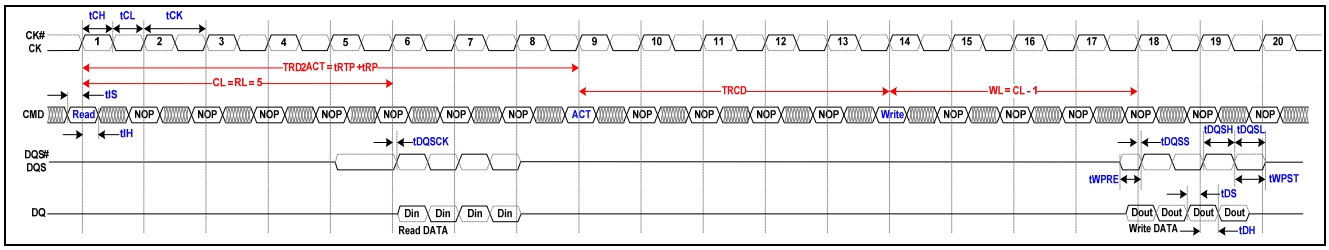


Figure 98. Douglas DDR2 Basic Read / Write Cycle Timing, CL = 5, Burst Length = 4

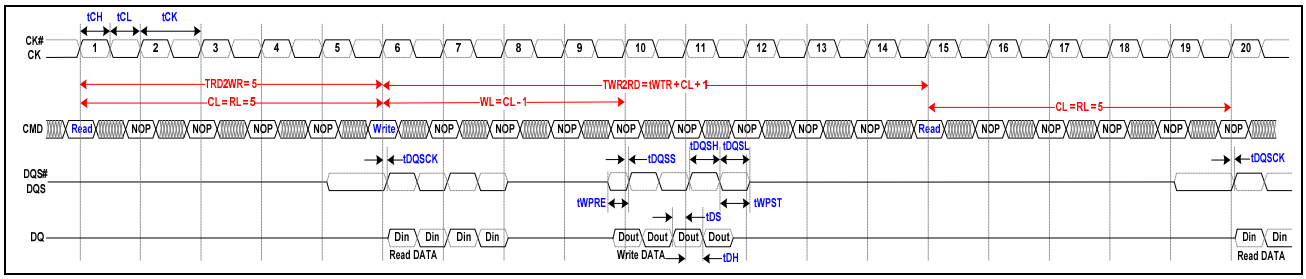


Figure 99. Douglas DDR2 Read-to-Write, Write-to-Read Turnaround Cycle Timing, CL = 5, Burst Length = 4

Table 51. Douglas DDR2 AC Timing For 333Mhz Operation

Parameter	Symbol	Min	Typ	Max	Units
Clock Period	tCK	3	3		ns
Clock High Width	tCH	1.35		1.65	ns
Clock Low Width	tCL	1.35		1.65	ns
Address and Control Input Hold Time	tIH	275	1200	-	ps
Address and Control Input Setup Time	tIS	200	700	-	ps
DQ and DM Input Setup Time	tDS	100	400	-	Ps
DQ and DM Input Hold Time	tDH	175	550	-	Ps
First DQS Latching to associated CLK	tDQSS	-0.75	-0.25	0.75	ns
DQS High Pulse Width	tDQSH	1.05		-	ns
DQS Low Pulse Width	tDQSL	1.05		-	ns
DQS Write Preamble	tWPRE	1.05		-	ns
DQS Write Postamble	tWPST	1.2		1.8	ns

The table below shows the recommended memory timing setup for most DDR2 DRAM devices @ 333Mhz with CAS latency = 5, which can be programmed through the FLI106xx registers MEMC_TIMING_1, MEMC_TIMING_2 and MEMC_TIMING_3.

Table 52. Recommended Memory Timing Setup For 333Mhz, CAS = 5 DRAM Devices

Symbol	Description	Min	Max	Suggested Value	Unit
CAS_LAT	CAS Latency	2	7	5	tCK
WR_LAT	Write Latency	1	7	4	tCK
TRC	Minimum delay of ACT to ACT within the same bank	0	31	20	tCK
TRRD	Minimum delay of ACT to ACT within different banks	1	7	4	tCK
TRCD	Min Latency of ACT to Read/Write within the same bank	1	7	5	tCK
TWR2RD	Minimum delay of Write to Read	0	15	9	tCK
TRD2WR	Minimum delay of Read to Write	0	15	5	tCK
TREFI	Refresh interval	0	32767	1599	tCK
TRFC	Auto-Refresh to ACT/Auto-Refresh period	0	255	35	tCK
TRD2ACT	Minimum delay of the last Read to ACT in different row of the same bank	0	31	8	tCK
TWR2ACT	Minimum delay of the last Write to ACT in different row of the same bank	0	31	16	tCK
TCKE	Minimum pulse width of CKE	3	3	3	tCK
TXSRD	Delay between Exit Self Refresh to a Read	200	200	200	tCK
TXP	Delay between Exit Precharge Power down to non-read CMD	2	2	2	tCK
TRP	Minimum delay of Precharge to ACT within the same bank	1	7	5	tCK

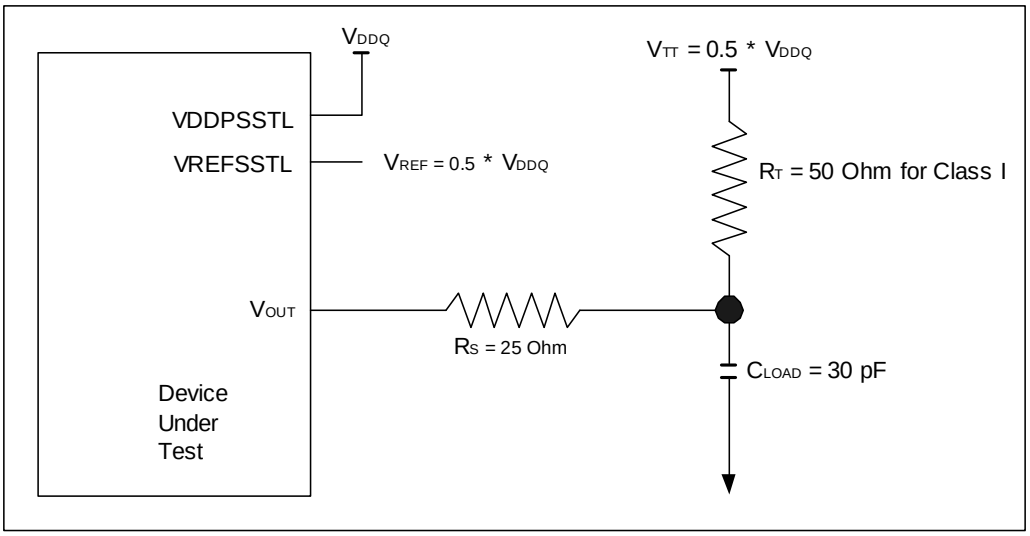


Figure 100. DDR I/F Test Load Circuit

Table 53. DDR I/F DC Specification (Threshold Voltage)

Parameter	Description	Min	Max	Unit
VIH (DC)	DC input logic high	DDR_VREF + 0.18	VDD_DDR + 0.3	V
VIL (DC)	DC input logic low	-0.3	DDR_VREF - 0.18	V
VIH (AC)	AC input logic high	DDR_VREF + 0.35		V
VIL (AC)	AC input logic low		DDR_VREF - 0.35	V

Table 54. DDR I/F DC Specification (Differential Voltage)

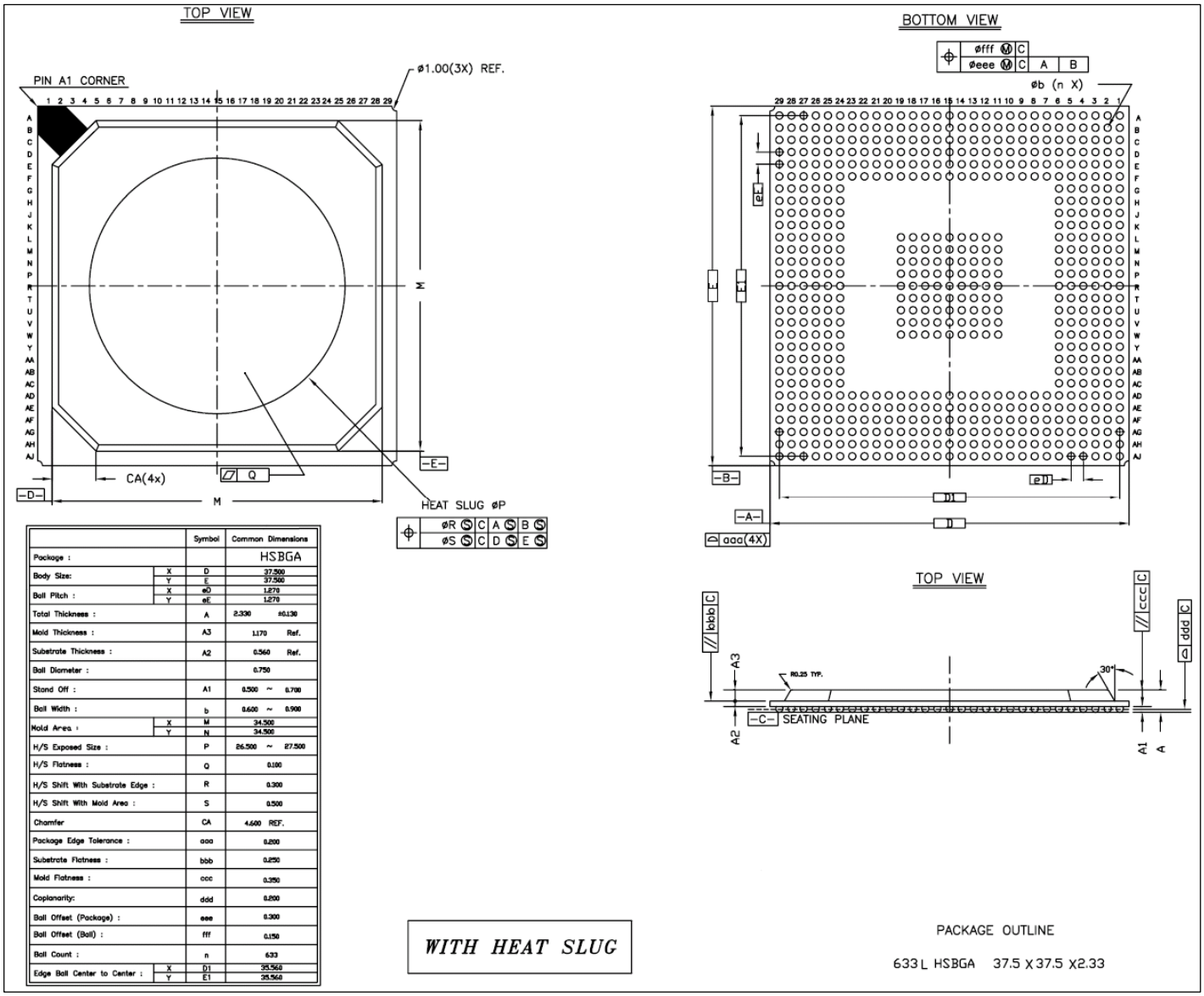
Parameter	Description	Min	Max	Unit
VIN (DC)	DC input signal voltage	-0.3	VDD_DDR+0.3	V
VID (DC)	DC differential input voltage	0.36	VDD_DDR + 0.6	V

Note: VID (AC) AC differential input voltage against VREF = 0.35V nominal.

30 PHYSICAL CHARACTERISTICS

The following physical characteristics apply to FLI106xx.

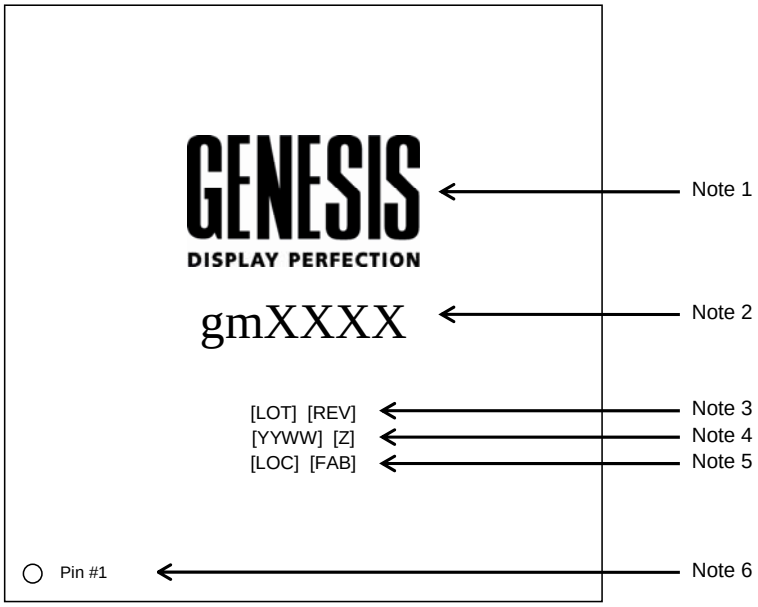
30.1 MECHANICAL DRAWING



		Symbol	Common Dimensions
Package :			HSBGA
Body Size:	X	D	37.500
	Y	E	37.500
Ball Pitch :	X	eD	1.270
	Y	eE	1.270
Total Thickness :		A	2.330 ±0.130
Mold Thickness :		A3	1.170 Ref.
Substrate Thickness :		A2	0.560 Ref.
Ball Diameter :			0.750
Stand Off :		A1	0.500 ~ 0.700
Ball Width :		b	0.600 ~ 0.900
Mold Area :	X	M	34.500
	Y	N	34.500
H/S Exposed Size :		P	26.500 ~ 27.500
H/S Flatness :		Q	0.100
H/S Shift With Substrate Edge :		R	0.300
H/S Shift With Mold Area :		S	0.500
Chamfer		CA	4.600 REF.
Package Edge Tolerance :		aaa	0.200
Substrate Flatness :		bbb	0.250
Mold Flatness :		ccc	0.350
Coplanarity:		ddd	0.200
Ball Offset (Package) :		eee	0.300
Ball Offset (Ball) :		fff	0.150
Ball Count :		n	633
Edge Ball Center to Center :	X	D1	35.560
	Y	E1	35.560

Figure 101. FLI106xx Mechanical Diagram

30.2 PACKAGE MARKING



Note 1: Genesis Logo

Must be prominently displayed

Note 2: Part Number

Font must be 2-3 sizes bigger than rest of text

gm – Genesis Microchip

FLI – Faroudja Laboratories (Genesis part that incorporates Faroudja technology)

XXXX – Alphanumeric part number

Note 3: Lot Code

[LOT] – Alphanumeric characters designation for lot number

[REV] – Two letter designation for mask ID revision

Note 4: Assembly Code

[YYWW] – YY = year; WW = workweek; when package is molded or sealed

[Z] – Assembly company

Note 5: Fab Location Code

[LOC] – Country of origin

[FAB] – Fab

Note 6: Pin #1 location identifier

Figure 102. FLI106xx Package Diagram

30.3 SOLDER REFLOW PROFILE

The following is the recommended solder reflow profile for Genesis Microchip lead-free BGA devices.

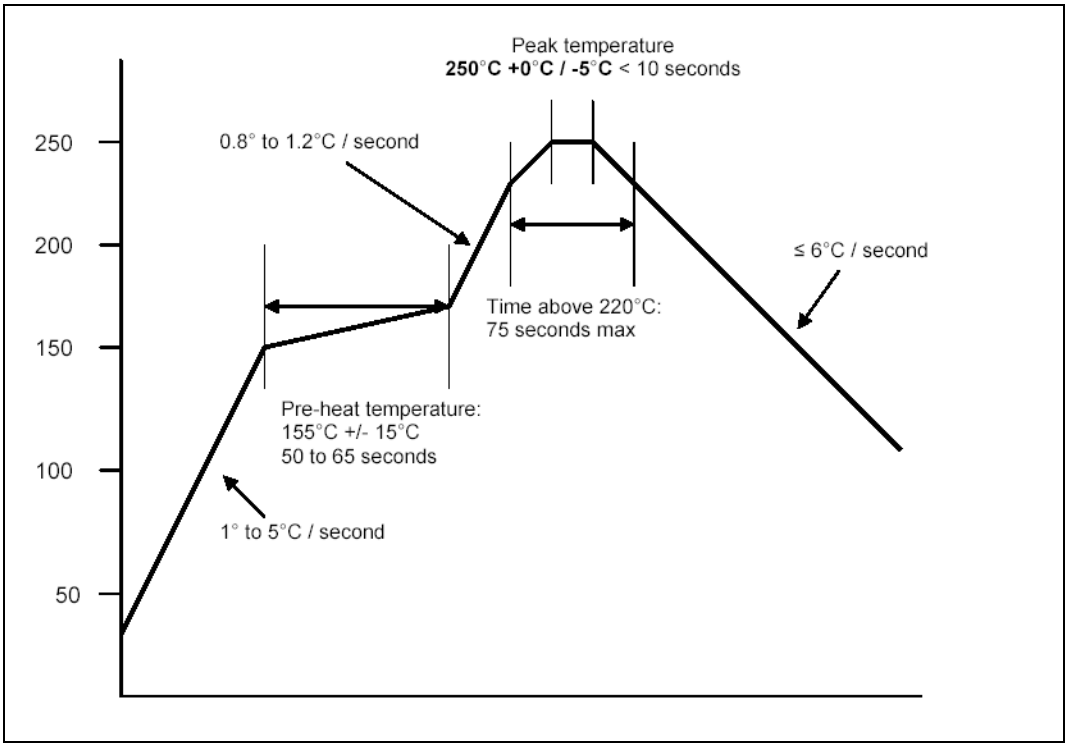


Figure 103. Lead-Free BGA Solder Reflow Profile

31 RoHS Compliance

Genesis Microchip Inc. has worked closely with its manufacturing subcontractors to eliminate banned substances under EU Directive 2002/95/EC on Restriction of Hazardous Substances (RoHS) from Genesis products. As of May 1st, 2006 (Assembly Date Code 0620), Genesis builds only RoHS compliant products, which do not exceed the following Maximum Concentration Values for the following banned substances under RoHS:

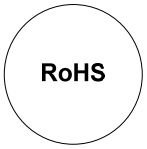
- Lead (Pb) 0.1% By Weight (or 1000 PPM)
- Hexavalent Chromium (Cr^{6+}) 0.1% By Weight (or 1000 PPM)
- Mercury (Hg) 0.1% By Weight (or 1000 PPM)
- Cadmium (Cd) 0.01% By Weight (or 100 PPM)
- Polybrominated Biethyls (PBB) 0.1% By Weight (or 1000 PPM)
- Polybrominated Diphenyl Ethers (PBDE) 0.1% By Weight (or 1000 PPM)

For our QFP (Quadra Flat Package) products, leads are plated using 100% Matte Tin. For our BGA (Ball Grid Array) products, the current qualified BGA solder ball compositions are:

- 95.5% Sn (Tin), 4.0% Ag (Silver), 0.5% Cu (Copper)
- 96.5% Sn (Tin), 3.0% Ag (Silver), 0.5% Cu (Copper)

The RoHS information provided above is, to our knowledge, correct as of the date of this Datasheet, based on information provided by third parties such as our manufacturing subcontractors. Genesis Microchip Inc. makes no representation or warranty as to the accuracy of such third party information. All of Genesis Microchip Inc. packaging assembly suppliers provide third party test certificates to ascertain compliance with RoHS requirements. These certificates, commonly referred to as ICP certificates, are provided to Genesis Microchip Inc. customers upon request and are indicative of the method of checking of materials used in our packages. Regarding Matte Tin Plating, Genesis Microchip Inc. packaging assembly suppliers provide Tin Whiskering reports for their plating lines on a regular basis, which are also available to Genesis Microchip Inc. customers upon request.

Genesis Microchip Inc. uses the below symbol for identification of RoHS Compliant products.



For any different requirements, please contact our Sales Department or a Product Manager.